DCON_DLL

User's Manual

(Version 3.5)

Dynamic Link Library (DLL) for DCON
(I-7000/8000/87K) Series Modules

Contents

1. INTRODUCTION

There are two software packages, which are DCON Utility and DCON_DLL, for ICPDAS DCON (I-7000/8000/87K) series modules. The **DCON Utility** is an utility/diagnostic package for WINDOWS 95/98/NT/2000/XP users. End-user can easily setup the configuration of I-7000/8000/87K series modules by this tool. The functions of DCON Utility are listed as below:

1. RS-232 COM port selection;
2. Search the DCON (I-7000, I-8000 and I-87K) series modules in control Network;
3. To setup the configuration for DCON series modules;
4. Module calibration;
5. Analog Input/Output;
6. Digital Input/Output;
7. Hi/Lo alarm setting;
8. Send Command string and receive response.

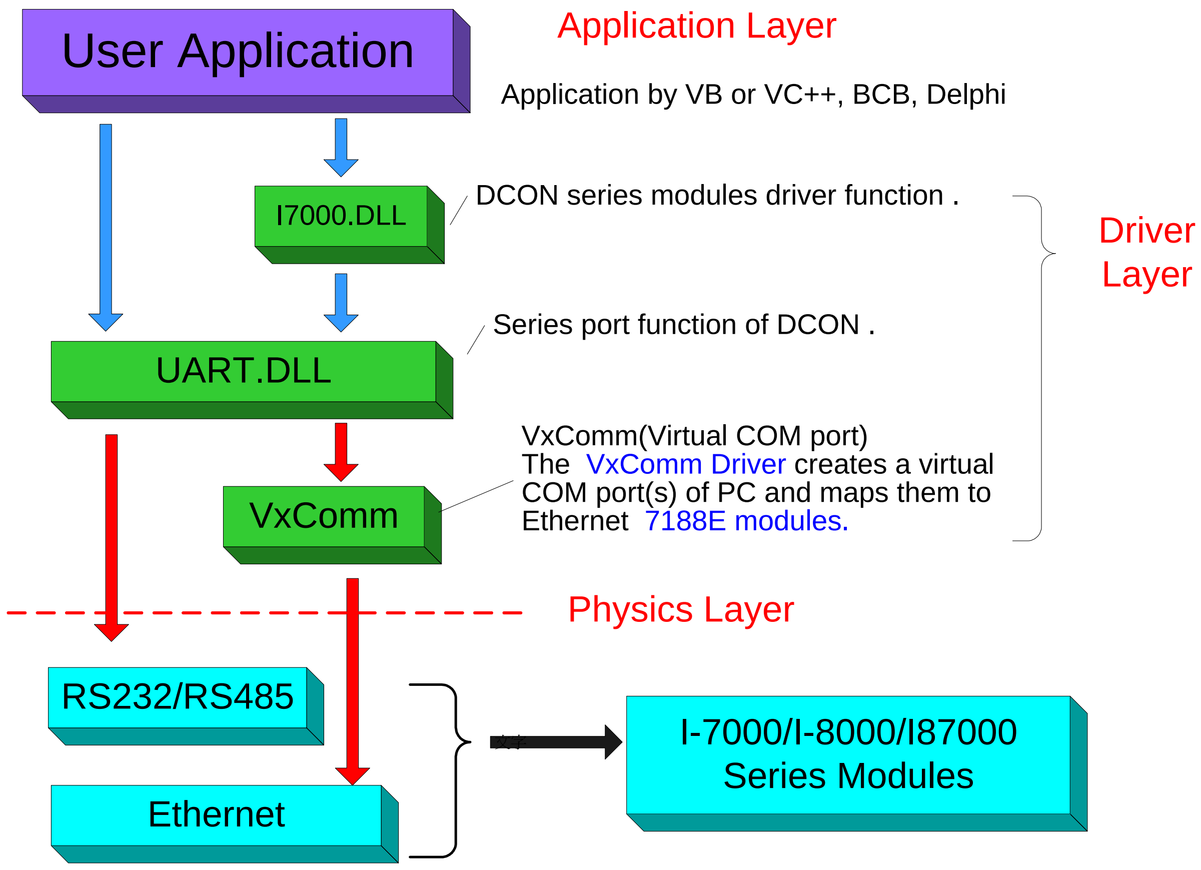
The **DCON_DLL** is a DLL functions designed for Windows 95/98/NT/2000/XP and can be utilized by VC^{++}, BC^{++}, VB, Delphi, and BC^{++} Builder. The features of DCON_DLL are given as following:

1. Provide general-purpose RS-232 application functions;
2. Provide general-purpose the command sending and response receiving functions of DCON (I-7000/8000/87K) series modules;
3. Provide high performance application functions for DCON series modules;
4. Provide several of demo programs for VC^{++}, VB, Delphi, BCB C^{++}.

2. ARCHITECTURES UNDER WINDOWS

The UART.DLL & I7000.DLL are the dynamic linking library (DLL) designed for Windows 95/98 and Windows NT 3.51/4.0/2000/XP applications. The users can apply it to develop their own application program through many programming languages such as VC^{++}, BC^{++}, BC^{++} Builder, VB, and Delphi. For your convenience application, there are many demo programs provided for VC^{++}, VB, Delphi and BC^{++} Builder. Based on the demo programs, User can easily understand how to use the function and develop their own application in a quick way.

The relationship among UART.DLL, I7000.DLL and user's application are depicted as follows:


User Application
Application Layer
Application by VB or VC++, BCB, Delphi
I7000.DLL
DCON series modules driver function .
Driver Layer
UART.DLL
Series port function of DCON .
VxComm
VxComm(Virtual COM port)
The VxComm Driver creates a virtual COM port(s) of PC and maps them to Ethernet 7188E modules.
Physics Layer
RS232/RS485
Ethernet
I-7000/I-8000/I87000 Series Modules

3. INSTALLATION

3.1 INSTALLATION STEPS

The following steps will demonstrate the installation process for the driver DCON_DLL. If users want to apply the DCON utility, please refer to DCON utility manual.

Step 1. Insert ICPDAS Product CD into CD ROM driver.

Step 2. Click Start/Run in the task bar.

Step 3. Enter the path as "CD Path:\Napdos\Driver\DCON_DLL\setup.exe".

Step 4. Follow the instructions to complete installation process.

Step 5. When the installation is finished, you can find all materials of the DCON driver are in the path "C:\DAQPro\DCON_DLL" and contains the following topics.

1. "What New" records the last information of DCON_DLL
2. "Manual folder" provides the manuals of DCON_DLL.
3. "Driver folder" contains all of library files, which are required in program development.
4. "Demo folder" includes the demo programs of DCON_DLL for different development environment.
5. "Demo Board folder" has the demo program and manual for Demo Board.

Note that the driver UART.DLL and I7000.DLL is copied to the directory C:\windows\system or C:\winnt\system32 (c:\windows\system32) for "Windows 95 or 98" or "Windows NT/2000/XP" system respectively. If users want to upgrade the DCON driver, it is allowed to download new release UART.DLL and I7000.DLL to this directory and replace these files.

The UART.DLL take care of the communication process of the PC's RS-232 port. For application, UART.H and UART.LIB stand as the header file and import library of UART.DLL respectively. The I7000.DLL is designed for the control application functions of I-7000/8000/87K series modules. That is, I7000.DLL will call UART.DLL to send command to and receive response from the DCON series modules.

3.2 README.TXT FILES

The DCON_DLL software will be continuously upgrade all the time. Therefore some information may not be given in this manual. All of the extra information will be provided in the files that are located in the companion floppy disk or CD-ROM as following after the installation.

\DCON_DLL\readme.txt	→ DCON_DLL release notes.
\DCON_DLL\WhatNew.txt	→ DCON_DLL release what's new.
\DCON_DLL\Demo\VC\DemoList.txt	→ The list of demo programs for VC^{++}.
\ DCON_DLL\Demo\VB\DemoList.txt	→ The list of demo programs for VB
\DCON_DLL\Demo\Delphi\DemoList.txt	→ The list of demo programs for Delphi
\DCON_DLL\Demo\BCB\DemoList.txt	→ The list of demo programs for BCB

The contents of readme.txt can be depicted as following:

- Release notes
- User's manual
- Demo program documentation
- Compiler & link documentation
- Application notes

It is recommended to read readme.txt file very carefully before starting to use this DCON software.

4. DLL APPLICATION

4.1 Application in VISUAL C^{++}

The whole demo programs of VC^{++} development enlivenment are given in directory "\DCON_DLL\Demo\Vc\". They are tested OK under Windows 95/98/NT/2000/XP and Visual C^{++} 4.0 development tool.

The VC^{++} user has to include these files as following:

1. \DCON_DLL\Driver\UART.DLL → functions to deal with RS-232
2. \DCON_DLL\Driver\I7000.DLL → functions for A/D, D/A, D/I, D/O
3. \DCON_DLL\ Driver \VC\UART.h →declarations for UART.DLL.
4. \DCON_DLL\ Driver \VC\I7000.h → declarations for I7000.DLL
5. \DCON_DLL\ Driver \VC\I8000.h →declarations for I7000. DLL
6. \DCON_DLL\ Driver \VC\I87000.h→declarations for I7000. DLL
7. \DCON_DLL\ Driver \VC\UART.lib→ import library of UART.DLL
8. \DCON_DLL\ Driver \VC\I7000.lib → import library of I7000. DLL
9. \DCON_DLL\ Driver \VC\I7000u.cpp→ functions for VC Demos.

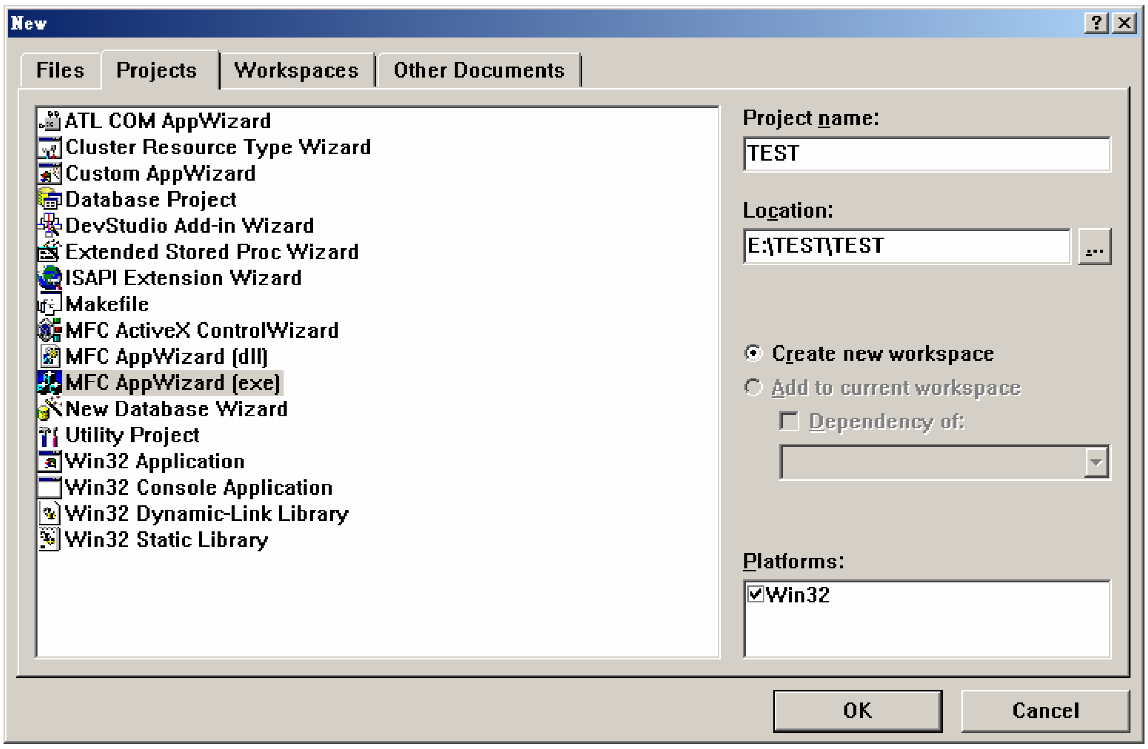
The key points for how to use these demo programs are given as following:

1. Enter the DOS command prompt under Windows.
2. Make sure the environment variable, PATH, which includes the Visual C^{++} compiler
3. Execute the \MSDEV\BIN\VCVARS32.BAT one time to setup the environment variable. Visual C++ will provide the "VCVARS32.BAT" file. The application program must include I7000.H, I8000.h, I87k and uart.h.
4. Copy the UART.LIB and I7000.LIB to the same directory with application program
5. Edit the demo program (refer to CON_DLL\Demo\vc \demoXX \ demoXX.C)
6. Edit the NMAKE file (Please refer to \DCON_DLL\Demo\vc\demoXX\demoXX. MAK)
7. Edit the BATCH file (refer to \DCON_DLL\Demo\vc\demoXX \c.bat)
8. Execute the batch file
9. Execute the execution file

4.2 Application in MFC

The usage of DCON_DLL in MFC application is very similar to the one in VC^{++}. The demo programs are tested OK under Windows 95/98/NT/XP and Visual C^{++} 4.0. The application key points are given in the following steps:

Step 1: Use MFC wizard to create source code



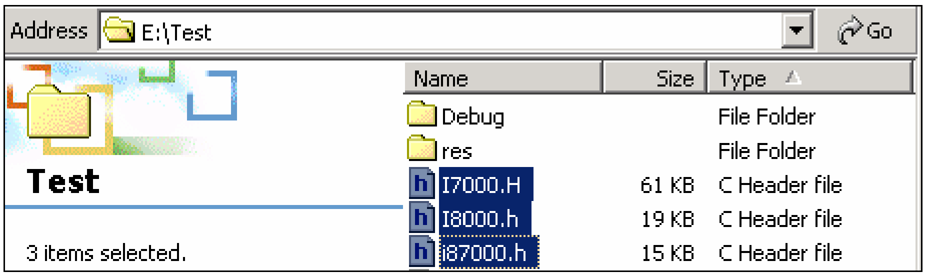
Project name: TEST

Location: E:\TEST\TEST

[✓ Create new workspace] Add to current workspace

Platforms: [✓] Win32

Step 2: The application program must include the head files shown in the below figure. The I7000.h and I8000.h and I87K.h are the declaration files for I-7000, I-8000 and I-87K series modules respectively.



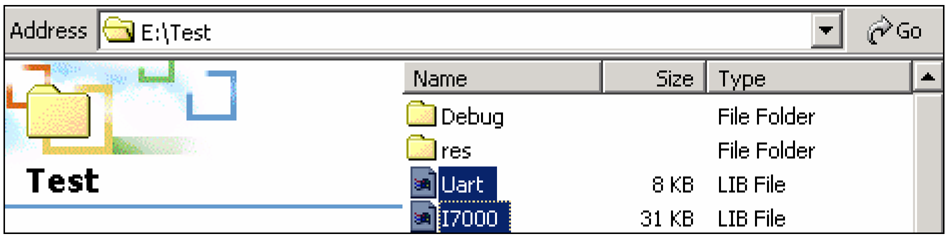
Address E:\Test

Name	Size	Type
Debug		File Folder
res		File Folder
I7000.H	61 KB	C Header file
I8000.h	19 KB	C Header file
i87000.h	15 KB	C Header file

Test

3 items selected.

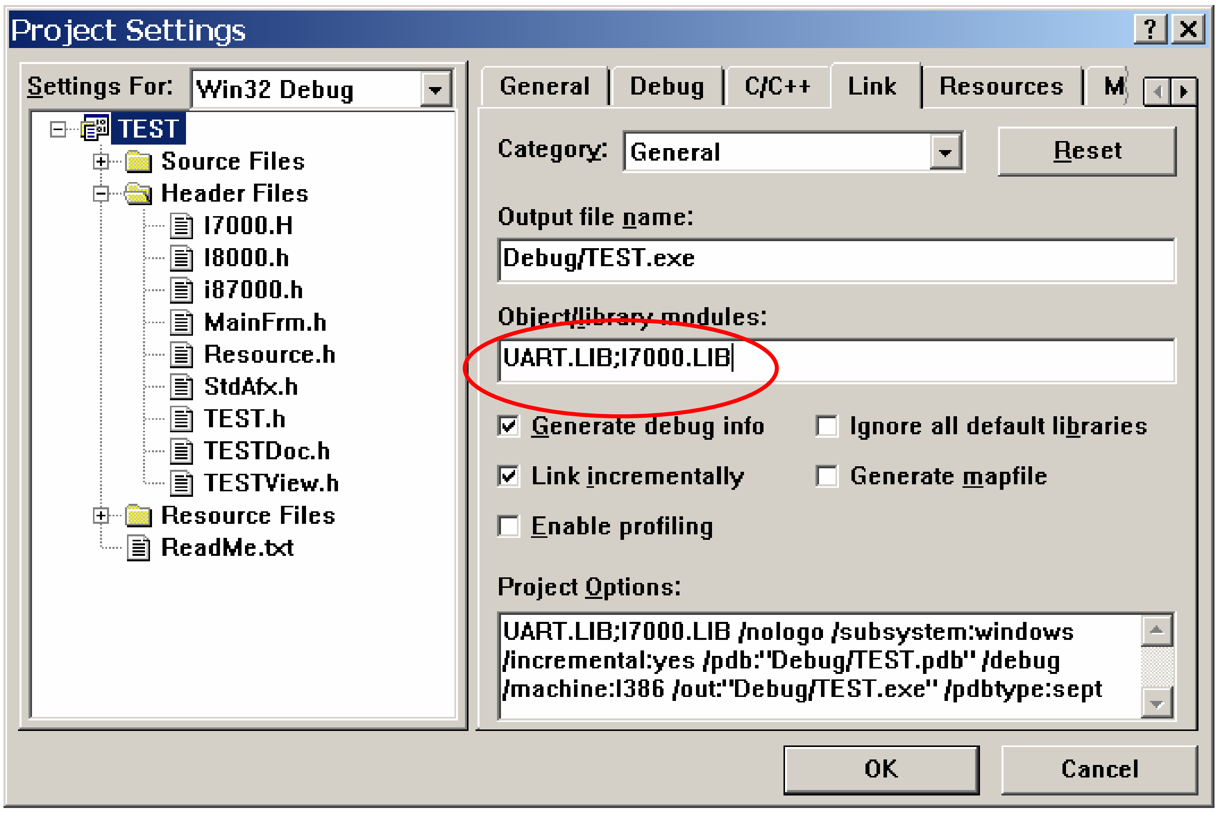
Step 3: Copy the UART.LIB and I7000.LIB to the same directory with application program.


Address E:\Test

Test

Name	Size	Type
Debug		File Folder
res		File Folder
Uart	8 KB	LIB File
I7000	31 KB	LIB File

Step 4: Select [Build/Settings/Link] and key in "UART.LIB" and "I7000.LIB" in the object/library modules field



Category: General

Output file name: Debug/TEST.exe

Object/library modules: UART.LIB;I7000.LIB

[✓] Generate debug info
[ ] Ignore all default libraries
[✓] Link incrementally
[ ] Generate mapfile
[ ] Enable profiling

Project Options:

```
UART.LIB;I7000.LIB /nologo /subsystem:windows
/incremental:yes /pdb:"Debug/TEST.pdb" /debug
/machine:I386 /out:"Debug/TEST.exe" /pdbtype:sept
```

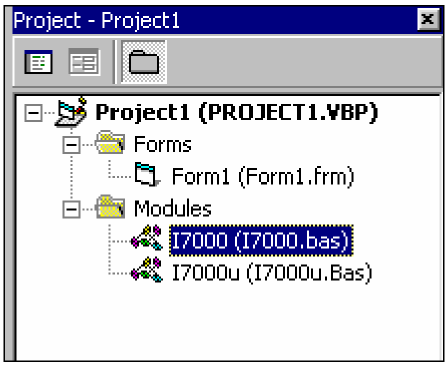
4.3 USING VISUAL BASIC

The demo programs are tested OK in Windows 95/98/NT/2000/XP and VB 5.0 version.

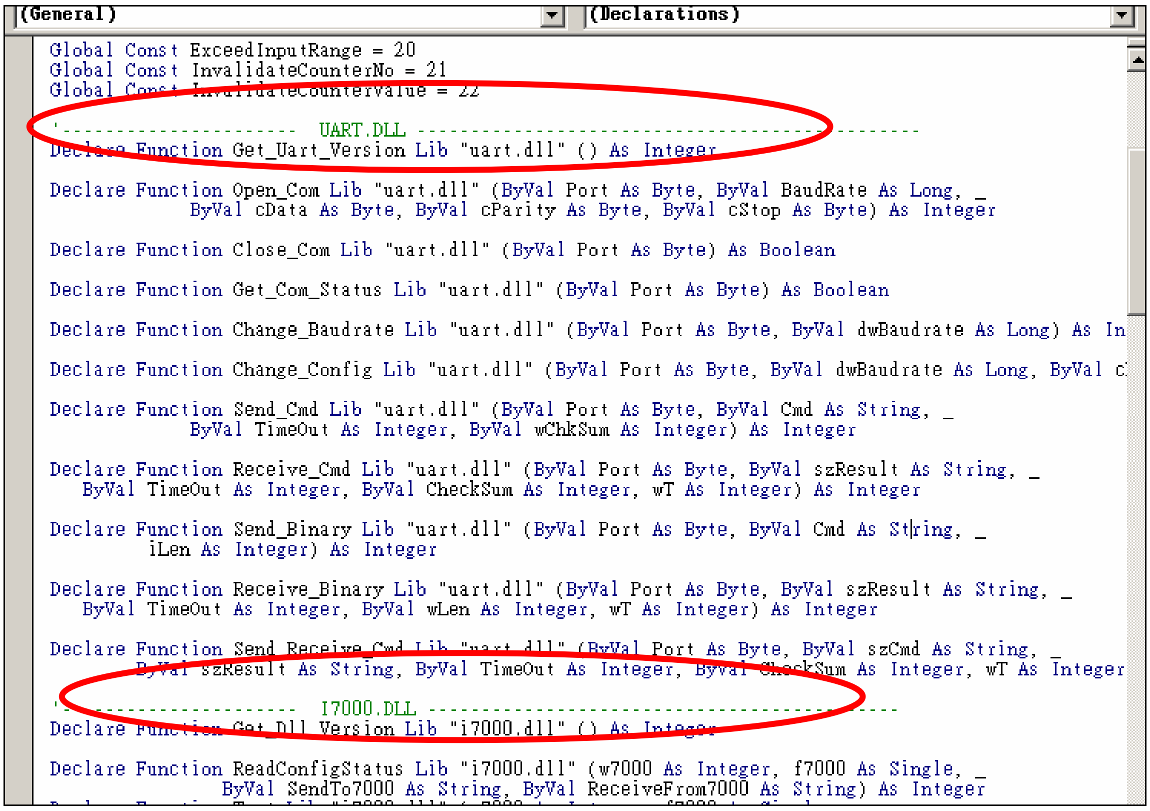
The user of VB appliocation has to implement these files as following:

1. \DCON_DLL\Driver\UART.DLL → functions to deal with RS-232
2. \DCON_DLL\Driver\I7000.DLL → functions for A/D, D/A, D/I, D/O
3. \DCON_DLL\ Driver \VB\I7000.bas → declarations for UART& I7000. DLL
4. \DCON_DLL\ Driver \VB\I7000u.bas→ some functions for VB Demos.

In the project files, users must include declaration files I-7000.bas and I7000u.bas into VB modules environment, as shown in below figure.



After double clicking on the I-7000.bas to open the file, users can see the declarations of function for UART.DLL and I7000.DLL and some defined constant declarations


(General)
(Declarations)

```
Global Const ExceedInputRange = 20
Global Const InvalidateCounterNo = 21
Global Const InvalidateCounterValue = 22

'----------------------  UART.DLL  -----------------------------------------------------
Declare Function Get_Uart_Version Lib "uart.dll" () As Integer

Declare Function Open_Com Lib "uart.dll" (ByVal Port As Byte, ByVal BaudRate As Long, _
             ByVal cData As Byte, ByVal cParity As Byte, ByVal cStop As Byte) As Integer

Declare Function Close_Com Lib "uart.dll" (ByVal Port As Byte) As Boolean

Declare Function Get_Com_Status Lib "uart.dll" (ByVal Port As Byte) As Boolean

Declare Function Change_Baudrate Lib "uart.dll" (ByVal Port As Byte, ByVal dwBaudrate As Long) As In

Declare Function Change_Config Lib "uart.dll" (ByVal Port As Byte, ByVal dwBaudrate As Long, ByVal c

Declare Function Send_Cmd Lib "uart.dll" (ByVal Port As Byte, ByVal Cmd As String, _
             ByVal TimeOut As Integer, ByVal wChkSum As Integer) As Integer

Declare Function Receive_Cmd Lib "uart.dll" (ByVal Port As Byte, ByVal szResult As String, _
   ByVal TimeOut As Integer, ByVal CheckSum As Integer, wT As Integer) As Integer

Declare Function Send_Binary Lib "uart.dll" (ByVal Port As Byte, ByVal Cmd As String, _
         iLen As Integer) As Integer

Declare Function Receive_Binary Lib "uart.dll" (ByVal Port As Byte, ByVal szResult As String, _
   ByVal TimeOut As Integer, ByVal wLen As Integer, wT As Integer) As Integer

Declare Function Send_Receive_Cmd Lib "uart.dll" (ByVal Port As Byte, ByVal szCmd As String, _
        ByVal szResult As String, ByVal TimeOut As Integer, ByVal CheckSum As Integer, wT As Integer

'----------------------  I7000.DLL  ----------------------------------------------------
Declare Function Get_Dll_Version Lib "i7000.dll" () As Integer

Declare Function ReadConfigStatus Lib "i7000.dll" (w7000 As Integer, f7000 As Single, _
              ByVal SendTo7000 As String, ByVal ReceiveFrom7000 As String) As Integer
```

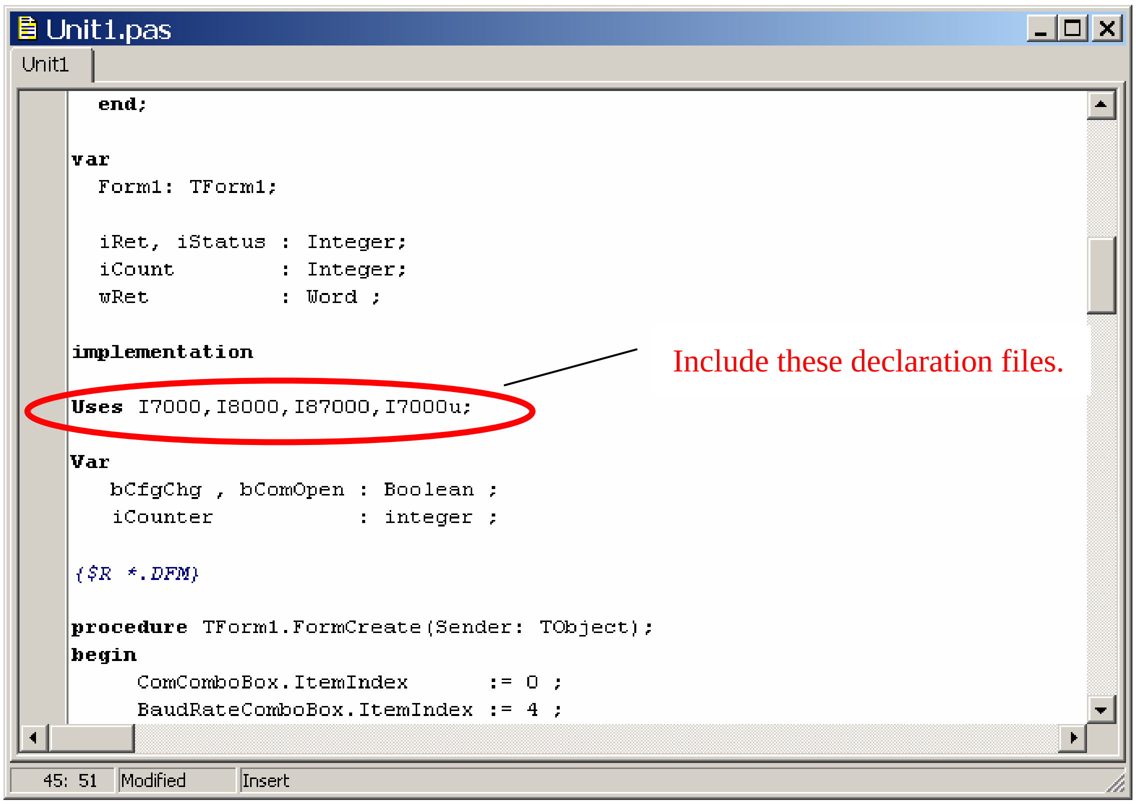
4.4 USING DELPHI

The demo programs are tested OK in Windows 95/98/NT/2000/XP and Delphi 3.0 version.

The Delphi user has to use these files as following:

1. \DCON_DLL\Driver\UART.DLL → functions to deal with RS-232
2. \DCON_DLL\Driver\I7000.DLL → functions for A/D, D/A, D/I, D/O
3. \DCON_DLL\ Driver \Delphi\I7000.pas → declarations for UART& I7000.DLL
4. \DCON_DLL\ Driver \Delphi\I8000.pas → declarations for I7000. DLL
5. \DCON_DLL\ Driver \Delphi\I87000.pas → declarations for I7000.DLL
6. \DCON_DLL\ Driver \Delphi\I7000u.pas → functions for Delphi Demos.

In the unit file, users must include declaration files I-7000.pas (I-8000.pas, I87k.pas) and I7000u.pas into Delphi environment, as shown in below figure. The I7000.pas, I8000.pas and I87k.pas are the declaration files for I-7000, I-8000 and I-87K series modules respectively.



```
  end;

var
  Form1: TForm1;

  iRet, iStatus : Integer;
  iCount        : Integer;
  wRet          : Word ;

implementation

Uses I7000,I8000,I87000,I7000u;

Var
   bCfgChg , bComOpen : Boolean ;
   iCounter           : integer ;

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
     ComComboBox.ItemIndex      := 0 ;
     BaudRateComboBox.ItemIndex := 4 ;
```

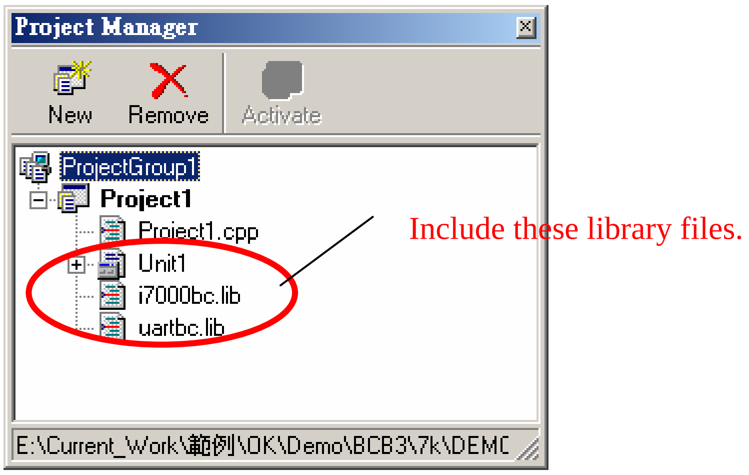
4.5 Application in Borland C^{++} Builder

The demo programs are tested OK in Windows 95/98/NT/2000/XP and C^{++} Builder 3.0 version.

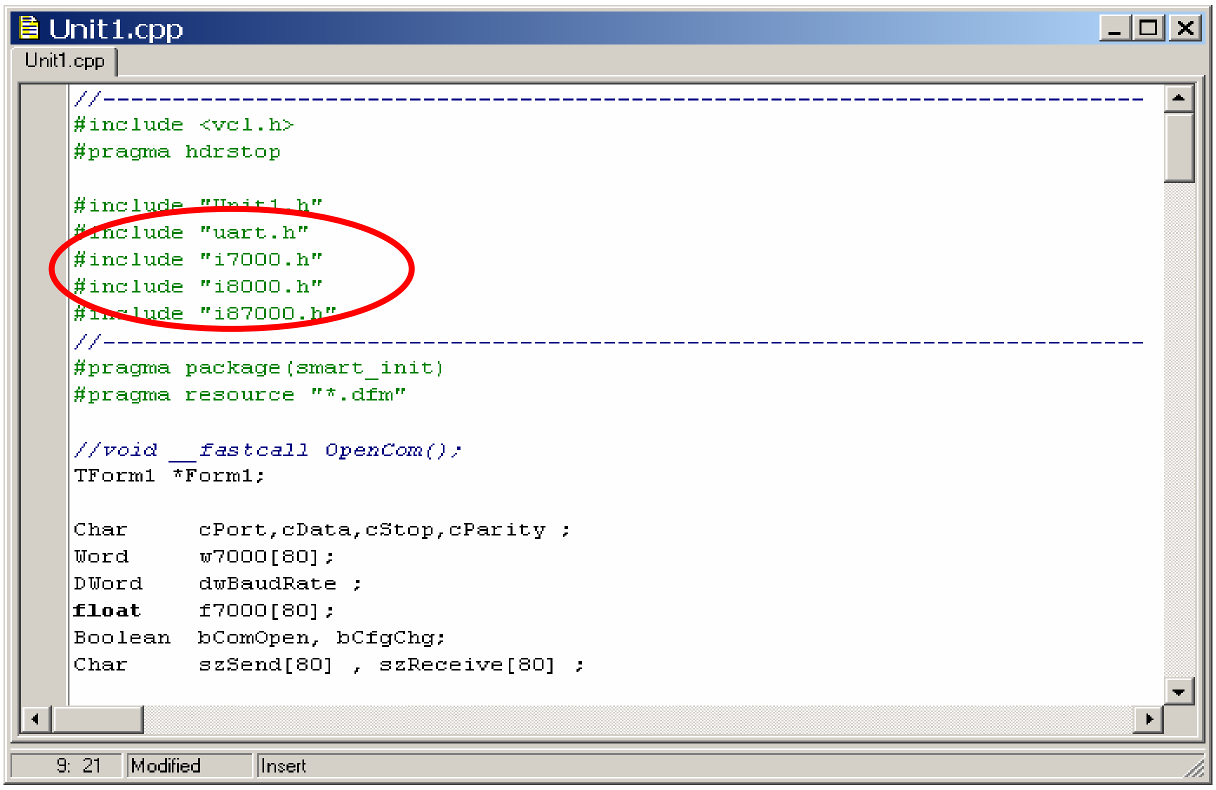
The BC^{++} & BC^{++} Builder user has to include these files as following:

1. \DCON_DLL\Driver\UART.DLL → functions to deal with RS-232
2. \DCON_DLL\Driver\I7000.DLL → functions for A/D, D/A, D/I, D/O
3. \DCON_DLL\ Driver \ BCB\uartbc.lib → import library of UART.DLL
4. \DCON_DLL\ Driver \BCB\I7000bc.lib → import library of I7000.DLL
5. \DCON_DLL\ Driver \BCB\UART.h → declarations for UART.DLL
6. \DCON_DLL\ Driver \BCB\I7000.h → declarations for I7000. DLL
7. \DCON_DLL\ Driver \BCB\I8000.h → declarations for I7000. DLL
8. \DCON_DLL\ Driver \BCB\I87000.h → declarations for I7000. DLL
9. \DCON_DLL\ Driver \BCB\I7000u.cpp → functions for BCB Demos.

The uartbc.lib and I7000bc.lib are different to the library files in application of VC^{++} and these two files must be included into the project, as shown in following figure.



In the unit1.cpp file, users must include declaration files I-7000.h (I-8000.h, I87k.h) and uart.h into BC^{++} builder environment, as shown in figure below. The I7000.h, I8000.h and I87K.h are the declaration files for I-7000, I-8000 and I-87K series modules respectively.


Unit1.cpp

Unit1.cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "uart.h"
#include "i7000.h"
#include "i8000.h"
#include "i87000.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"

//void __fastcall OpenCom();
TForm1 *Form1;

Char      cPort,cData,cStop,cParity ;
Word      w7000[80];
DWord     dwBaudRate ;
float     f7000[80];
Boolean   bComOpen, bCfgChg;
Char      szSend[80] , szReceive[80] ;
```

9: 21 Modified Insert

5. UARTDLL

■ Open_Com

Description:

This function is used to configure and open the COM port. It must be **called once before** sending/receiving command through COM port.

Syntax:

Open_Com(char cPort, DWORD dwBaudRate, char cData, char cParity, char cStop)

Input Parameter:

cPort:	1=COM1, 2=COM2 …. , 255=COM255
dwBaudRate:	50/75/110/134.5/150/300/600/1200/1800/2400/4800/7200/ 9600/19200/38400/57600/115200
cData:	5/6/7/8 data bit
cParity:	0=NonParity, 1=OddParity, 2=EvenParity
cStop:	0=1-stop, 1=1.5-stp, 2=2-stop

NOTE cData=8, cParity=0, cStop=0 is the default for DCON series modules.

Return Value:

NoError :	OK
Others :	Error code

NOTE About the "Error Code" please refer to "Appendix C Error Code".

■ Close_Com

Description:

This function closes and releases the resources of the COM port from computer recourse. And it must be **called before exiting the application program**. The Open_Com will return error message if the program exit without calling Close_Com function.

Syntax:

```
Close_Com(char cPort)
```

Input Parameter:

cPort :	1=COM1, 2=COM2 …. , 255=COM25

Return Value:

NoError :	OK
Others :	Error code.

■ Send_Receive_Cmd

Description:

This function sends a command string to RS485 Network and receives the response from RS485 Network. If the wCheckSum=1, this function automatically adds the two checksum bytes into the command string and also check the checksum status when receiving response from the modules. Note that the end of sending string is added [0x0D] to mean the termination of every command. This Send_Receive_Cmd is not a multi-task DLL.

Syntax:

Send_Receive_Cmd (char cPort, char szCmd[], char szResult[], WORD wTimeOut, WORD wCheckSum, WORD *wT)

Input Parameter:

cPort:	1=COM1, 2=COM2, 3=COM3, 4=COM4 …. , 255=COM255
szCmd:	Sending command string
szResult:	Receiving the response string from the modules
wTimeOut:	Communicating timeout setting, time unit = 1ms
wCheckSum:	0=DISABLE, 1=ENABLE
*wT:	Total time of send/receive interval, unit = 1 ms

Return Value:

NoError :	OK
Others :	Error code

■ Send_Cmd

Description:

This function only sends a command string to DCON series modules. If the wCheckSum=1, it automatically **adds the two checksum bytes to the command string**. And then the end of sending string is further added [0x0D] to mean the termination of the command (szCmd). Note that the function Send_Cmd is a multi-task and multi-thread DLL. And this command string cannot include space char within the command string. Otherwise, the command string will be stoped by space character. For example:"$01M 02 03" is user's command string. However, the actual command sent out is "$01M".

Syntax:

Send_Cmd (char cPort, char szCmd[], WORD wTimeOut, WORD wCheckSum)

Input Parameter:

cPort:	1=COM1, 2=COM2, 3=COM3, 4=COM4, …. , 255=COM255
szCmd:	Sending command string(Terminated with "0")
wTimeOut:	Communicating timeout setting, time unit = 1ms
wCheckSum:	0=DISABLE, 1=ENABLE

Return Value:

NoError :	OK
Others :	Error code

■ Receive_Cmd

Description:

Users can utilize this function to obtain the response string from the modules in RS-485 Network. And this function provides a response string without the last byte [0x0D].

Syntax:

Receive_Cmd (unsigned char cPort, char szResult[], WORD wTimeOut, WORD wChksum, WORD *wT);

Input Parameter:

Cport::	1=COM1, 2=COM2, 3=COM3, 4=COM4,… 255=COM255
szResult:	Receiving the response string from the modules
wTimeOut::	Communicating timeout setting, time unit = 1ms
wCheckSum:	0=DISABLE, 1=ENABLE
*wT:	Total time of receiving interval, unit = 1 ms.

Return Value:

NoError :	OK
Others :	Error code

■ Send_Binary

Description:

Send out the command string by fix length, which is controlled by the parameter "iLen". The difference between this function and Send_cmd is that Send_Binary terminates the sending process by the string length "iLen" instead of the character "CR"(Carry return). Therefore, this function can send out command string with or without null character under the consideration of the command length. Besides, because of this function without any error checking mechanism (Checksum, CRC, LRC... etc.), users have to add the error checking information to the raw data by themselves if communication checking system is required. Note that this function is usually applied to communicate with the other device, but not for ICPDAS DCON (I-7000/8000/87K) series modules.

Syntax:

Send_Binary (unsigned char cPort, char szCmd[], int iLen);

Input Parameter:

cPort:	1=COM1, 2=COM2, 3=COM3, 4=COM4, …, 255=COM255
szCmd:	Sending command string(Terminated with "0")
ILen :	The length of command string.

Return Value:

NoError :	OK
Others :	Error code

■ Receive_Binary

Description:

This function is applied to receive the fix length response. The length of the receiving response is controlled by the parameter “iLen”. The difference between this function and Receive_cmd is that Receive_Binary terminates the receiving process by the string length “iLen” instead of the character "CR"(Carry return). Therefore, this function can be used to receive the response string data with or without null character under the consideration of receiving length. Besides, because of this function without any error checking mechanism (checksum, CRC, LRC... etc.), users have to remove the error checking information from the raw data by themselves if communication checking system is used. Note that this function is usually applied to communicate with the other device, but not for ICPDAS DCON (I-7000/8000/87K) series modules.

Syntax:

Receive_Binary (unsigned char cPort, char szResult[], WORD wTimeOut, WORD wLen, WORD *wT);

Input Parameter:

cPort:	1=COM1, 2=COM2, 3=COM3, 4=COM4, …, 255=COM255
szResult:	The receiving string from the module
wTimeOut:	Communicating timeout setting, time unit = 1ms
wLen:	The length of result string.
*wT:	Total time of receiving interval, unit = 1 ms.

Return Value:

NoError :	OK
Others :	Error code

■ Get_Com_Status

Description:

The function can obtain COM Port status. If return value is “0” (false), it means “**The COM Port is not in used!**”. Otherwise, if the return value is “1” (true), it means “**The COM Port is in used!**”

Syntax:

Get_Com_Status (char cPort)

Input Parameter:

cPort:	1=COM1, 2=COM2, 3=COM3, 4=COM4, …, 255=COM255

Return Value:

0 :	COM port is **not in used.**
1 :	COM port is **in used.**

■ Change_BaudRate

Description:

This function only can be applied to change the Baudrate setting of serial communication after COM port was opened.

Syntax:

Change_BaudRate (char cPort, DWORD dwBaudrate)

Input Parameter:

cPort :	1=COM1, 2=COM2, 3=COM3, 4=COM4, …, 255=COM255
dwBaudrate:	50/75/110/134.5/150/300/600/1200/1800/2400/4800/7200 9600/19200/38400/57600/115200

Return Value:

NoError :	OK
Others :	Error code

■ Change_Config

Description:

This function only can be used to change the configuration of the COM port after COM port was opened.

Syntax:

Change_Config (char cPort, DWORD dwBaudRate, char cData, char cParity, char cStop)

Input Parameter:

cPort:	1=COM1, 2=COM2 …. , 255=COM255
dwBaudRate:	50/75/110/134.5/150/300/600/1200/1800/2400/4800/7200 /9600/19200/38400/57600/115200
cData:	5/6/7/8 data bit
cParity:	0=NonParity, 1=OddParity, 2=EvenParity
cStop:	0=1-stop, 1=1.5-stp, 2=2-stop

Return Value:

NoError :	OK
Others :	Error code

■ Get_Uart_Version

Description:

Users can obtain the version information of UART.DLL.

Syntax:

Get_Uart_version (void)

Input Parameter:

Users don’t need to provide any parameter.

Return Value:

Return the version of UART.DLL with hexadecmal format.

Demo Program:

```
int ver ;              //Set “ver” is a integer Variable
ver=Get_Uart_Version();
// If the return value is 0X202, it means “the version of UART.DLL” is 2.0.2
```

6. I7000.DLL

The functions of I7000.DLL can be clarified as 8 groups as depicted as below:

1. Analog Input Functions;
2. Module Alarm Functions ;
3. Stain Gauge module Functions;
4. Analog Output Functions;
5. Digital Input Functions;
6. Digital Output Functions;
7. Counter module Functions;
8. Dual WatchDog and safety Functions.

The following sections will explore the more detail description for all functions.

■ Get_Dll_Version

Description:

Obtain the version information of I7000.DLL.

Syntax:

Get_Dll_Version (void)

Input Parameter:

Users don't need to provide any parameter.

Return Value:

Return the version message by hexadecmal format.

Demo Program:

```
int ver ;                //define "ver" is a integer Variable
ver=Get_DLL_Version();
// If the return value is 0X505, it means "the version of I7000.DLL" is 5.0.5
```

6.1 Analog Input Functions

6.1.1 I-7000 series modules

■ AnalogIn

Description:

Obtain the analog input value from DCON series modules.

Syntax:

AnalogIn (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7013/7014/7017/7018/7033
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	Channel number for 7017/7018/7033
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

Note: **"wBuf[6]" is the debug setting. If this parameter is set as "1", users can get whole command string and result string from szSend[ ] and szReceive[ ] respectively.**

fBuf: Float Input/Output Table

fBuf[0] :	**Analog input value return**

■ AnalogInHex

Description:

Obtain the analog input value in “Hexadecimal” format from I-7000 series modules.

Syntax:

AnalogInHex (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7013/7014/7017/7018/7033
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	Channel number for 7017/7018/7033
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**The analog input value in “Hexadecimal” format.**

fBuf: Float Input/Output Table

Not used.

Note: **Users have to use DCON utility to set up the analog input configuration of the module in hex format.**

■ AnalogInFsr

Description:

Obtain the analog input value in “FSR” format from I-7000 series modules. The “FSR” means “Percent” format.

Syntax:

AnalogInFsr (WORD wBuf [], float fBuf [], char szSend [], char szReceive [])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7013/7014/7017/7018/7033
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	Channel number for 7017/7018/7033
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**The analog input value.**

Note : Users have to use DCON utility to set up the analog input configuration of the module in FSR format.

■ AnalogInAll

Description:

Obtain analog input values of all channels from I-7017 or I-7018 or I-7033.

Syntax:

AnalogInAll (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]	COM port number: 1/2/3/4..../255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]	Module ID: 0x7017/18/33
wBuf[3]	Checksum: 0=disable, 1=enable
wBuf[4]	Time out setting, normal=100, unit=ms.
wBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]	**Analog input value of channel_0**
fBuf[1	**Analog input value of channel_1**
…………………..	
fBuf[7]	**Analog input value of channel_7**

■ ThermocoupleOpen_7011

Description:

This function can be used to detect the thermocouple state of I-7011 module for the supporting type “J, K, T, E, R, S, B, N, C” is open or close. If the response value is “0”, thermocouple I-7011 is working in close state. If the response value is “1”, thermocouple I-7011 is working in open state. For more information please refer to user manual.

Syntax:

ThermocoupleOpen_7011(WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[5] :	**0 → the thermocouple is close**
	1 → the thermocouple is open

fBuf: Float Input/Output Table

Not used

■ SetLedDisplay

Description:

Configure LED Display for specified channel of I-7033 or I-7016.

Syntax:

SetLedDisplay (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID; 0X7033, 0X7016;
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Set display channel for 7033 or 7016
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ GetLedDisplay

Description:

Get the current setting of the specified channel for LED Display channel in I-7033 or I-7016.

Syntax:

GetLedDisplay (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID 0x7033, 0x7016.
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**Current channel for LED display.**
	0=channel_0; 1=channel_1
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

6.1.2 I-8000 series modules

■ AnalogIn_8K

Description:

Obtain the analog input value in float format from I-8000 series modules.

Syntax:

AnalogIn_8K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8017h/0x8018/0x8013/0x8033
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The channel number of analog input module for 8013/17/18
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	**Slot number; the I/O module installed in I-8000 main unit.**

fBuf: Float Input/Output Table

fBuf[0] :	**Analog input value**

■ AnalogInHex_8K

Description:

Obtain the analog input value in “Hexadecimal” format from I-8000 series modules.

Syntax:

AnalogInHex_8K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8013/0x8017h/0x8018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The channel number of analog input module for 8013/17/18
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive.
dwBuf[7] :	**Slot number; the I/O module installed in I-8000 main unit.**
dwBuf[8]:	**The analog input value in Hex format**.

fBuf: Float Input/Output Table

Not used.

■ AnalogInFsr_8K

Description:

Obtain the analog input value in "FSR" format from I-8000 series modules. The "FSR" means "Percent" format.

Syntax:

AnalogInFsr_8K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8017/0x8018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The channel number of analog input module for 8013/17/18
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive.
dwBuf[7] :	**Slot number; the I/O module installed in I-8000 main unit.**

fBuf: Float Input/Output Table

fBuf[0]:	**The analog input value.**

■ AnalogInAll_8K

Description:

Obtain the analog input values of all channels from I-8013, I-8017 or I-8018.

Syntax:

AnalogInAll_8K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8013/8017/8018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	**Slot number; the I/O module installed in I-8000 main unit.**

fBuf: Float Input/Output Table

fBuf[0]	**analog input value of channel_0**
fBuf[1]	**analog input value of channel_1**
……………………	
fBuf[7]	**analog input value of channel_7**

6.1.3 I-87K series modules

■ AnalogIn_87K

Description:

Obtain the analog input value from I-87K series analog input modules, for example: I-87017 or I-87018.

Syntax:

AnalogIn_87K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87013/0x87017/0x87018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	Channel number of analog input module for 87013/17/18/33
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive.

fBuf: Float Input/Output Table

fBuf[0] :	**The analog input value return**

■ AnalogInHex_87K

Description:

Obtain the analog input value in “Hexadecimal” format from I-87K series analog input modules.

Syntax:

AnalogInHex_87K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87013/87017/87018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The channel number of analog input module for 87013/17/18
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7]:	**The analog input value in “Hex” format**.

fBuf: Float Input/Output Table

Not used.

■ AnalogInFsr_87K

Description:

Obtain the analog input value in “FSR” format from I-87K series analog input modules.

Syntax:

AnalogInFsr_87K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87013/87017/87018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The channel number of analog input module for 87013/17/18
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**The analog input value.**

■ AnalogInAll_87K

Description:

Obtain the analog input values of all channels from I-87013, I-87017 and I-87018.

Syntax:

AnalogInAll_87K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87013/0x87017/87018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]	**analog input value of channel_0**
fBuf[1]	**analog input value of channel_1**
…………………..	
fBuf[7]	**analog input value of channel_7**

6.2 Module Alarm Functions

6.2.1 I-7000 series modules

■ EnableAlarm

Description:

Enable the alarm function of I-7000 series modules and configure it in the status of momentary alarm and latch alarm mode. This fuction currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

EnabledAlarm (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7014/7016
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → momentary alarm mode
	1 → latch alarm mode
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DisableAlarm

Description:

Disable alarm function of I-7000 series modules. This fuction currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

DisableAlarm (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7014/7016
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5]	Not used
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ClearLatchAlarm

Description:

This function can clear the alarm latched of I-7000 series modules. This fuction currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

ClearLatchAlarm (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7011/7012/7014,/7016
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5]	Not used
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ SetAlarmLimitValue

Description:

Set the high or low alarm limit value of I-7000 series modules. This function currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

SetAlarmLimitValue (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011/0x7012/0x7014/0x7016
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → low alarm value setting
	1 → high alarm value setting
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0] :	Alarm value

■ ReadAlarmLimitValue

Description:

Obtain the high or low alarm limit value of I-7000 series modules. This fuction currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

ReadAlarmLimitValue (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011/0x7012/0x7014/7016.
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → low alarm value setting
	1 → high alarm value setting
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0] :	**Alarm value**

■ ReadOutputAlarmState

Description:

Obtain the alarm mode and alarm digital output value of I-7000 series modules. This fuction currently supports I-7011, I-7012, I-7014, and I -7016.

Syntax:

ReadOutputAlarmState (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011/0x7012/0x7014/0x7016.
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	No used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**0 → alarm disable**
	1 → momentary alarm
	2 → latch alarm
wBuf[8]:	**0 → DO:0 off DO:1 off**
	1 → DO:0 on DO:1 off
	2 → DO:0 off DO:1 on
	3 → DO:0 on DO:1 on

fBuf: Float Input/Output Table

Not used

6.2.2 I-8000 series modules

■ SetAlarmMode_8K

Description:

Disable or enable the alarm function of I-8000 series modules into momentary alarm or latch alarm mode. This function currently supports I-8013, I-8017h, I-8018, and I-8033.

Syntax:

SetAlarmMode_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8013/8017/8018/8033
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The specified channel number for I-8013/8017/8018/8033
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive
dwBuf[7] :	**Slot number; the I/O module installed in I-8000 main unit.**
dwBuf[8] :	0 → Low Alarm
	1 → High Alarm
dwBuf[9] :	0 → disable
	1 → Momentary alarm mode
	2 →Latch alarm mode

fBuf: Float Input/Output Table

Not used

■ SetAlarmConnect_8K

Description:

This function makes a connection effect between DO module and alarm function of analog input modules in I-8000 main unit. That is, if the alarm function of analog input has happened, then the specified DO channel of DO modules produce the defined output.

Syntax:

SetAlarmConnect_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8013/8017/8018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5]	The specified channel number for I-8013/8017/8018/8033
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.;
dwBuf[8] :	0 → Low Alarm
	1 → High Alarm
dwBuf[9] :	The slot number of DO module .
dwBuf[10] :	The defined DO channel according to the alarm function

fBuf: Float Input/Output Table

Not used

■ ClearLatchAlarm_8K

Description:

Clear the high or low latch alarm of analog input modules for I-8000 series modules.

Syntax:

ClearLatchAlarm_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8013/8017/8018
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The defined analog input channel No.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.
dwBuf[8] :	0 → Low Alarm
	1 → High Alarm

fBuf: Float Input/Output Table

Not used

■ SetAlarmLimitValue_8K

Description:

Configure the high or low alarm limit value of analog input modules for I-8000 series modules.

Syntax:

SetAlarmLimitValue_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2]:	Module ID, 0x8013/8017/8018
dwBuf[3]:	0=checksum disable, 1=checksum enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The defined analog input channel No.
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.
dwBuf[8]:	0→ Set low alarm value
	1→ Set high alarm value

fBuf: Float Input/Output Table

fBuf[0] :	Alarm value

■ ReadAlarmLimitValue_8K

Description:

Obtain the high or low alarm limit value of analog input modules for I-8000 series modules.

Syntax:

ReadAlarmLimitValue_8K(DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to 8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2]:	Module ID, 0x8013/8017/8018/8033
dwBuf[3]:	0=checksum disable, 1=checksum enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The defined analog input channel No.
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.
dwBuf[8]:	0 → Set low alarm value
	1 → Set high alarm value

fBuf: Float Input/Output Table

fBuf[0] :	**Alarm value**

■ ReadAlarmMode_8K

Description:

Obtain the alarm mode setting of analog input modules for I-8000 series modules.

Syntax:

ReadAlarmMode_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2]:	Module ID, 0x8013/8017/8018
dwBuf[3]:	0=checksum disable, 1=checksum enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The defined analog input channel No.
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7]:	Slot number; the I/O module installed in I-8000 main unit.
dwBuf[8]:	0→ Read low alarm mode setting
	1→ Read high alarm mode setting
dwBuf[9]:	**0 → Alarm disable**
	1 → Momentary alarm
	2 → Latch alarm

fBuf: Float Input/Output Table

Not used

■ ReadAlarmStatus_8K

Description:

Obtain the alarm status of analog input modules for I-8000 series modules.

Syntax:

ReadAlarmStatus_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Ohers:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0];	COM port number: 1 to 255	
dwBuf[1];	Module address: 0x00 to 0xFF	
dwBuf[2];	Module ID: 0x8013/17/18/33	
dwBuf[3];	Checksum: 0=disable, 1=enable	
dwBuf[4];	Time out setting, normal=100, unit=ms.	
dwBuf[5];	The defined analog input channel No.	
dwBuf[6];	0 → no save to szSend & szReceive	
	1 → save to szSend & szReceive	
dwBuf[7];	Slot number; the I/O module installed in I-8000 main unit.	
dwBuf[8]	**1: High Alarm Occur**	**0: Don't Occur**
dwBuf[9]	**1: Low Alarm Occur**	**0: Don't Occur**

fBuf: Float Input/Output Table

Not used

6.3 Strain Gauge Functions

■ SetupLinearMapping

Description:

Configure the linear mapping translation of I-7014 or I-7016 module from raw data range to target range data value. However, before using this function user need to get the module's range code of the module by calling “ReadConfigStatus () and set it into wBuf [7]. That is, this function provides linear mapping from range area [a, b] to [c, d], where fBuf[0]=a, fBuf[1]=b, fBuf[2]=c, fBuf[3]=d.

Syntax:

SetupLinearMapping (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7014, 0x7016
wBuf[3]	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	Flag: 0=no save, 1=save send/receive string
wBuf[7]:	Range code of this module

fBuf: Float Input/Output Table

fBuf[0]:	Source low value,	a
fBuf [1]:	Source high value,	b
fBuf [2]:	Target low value,	c
fBuf [3]:	Target high value,	d

■ EnableLinearMapping

Description:

Enable linear mapping function for I-7014 or I-7016 module.

Syntax:

EnableLinearMapping (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]	COM port number, 1 to 255
wBuf[1]	Module address: 0x00 to 0xFF
wBuf[2]	Module ID: 0x7014, 0x7016
wBuf[3]	Checksum: 0=disable, 1=enable
wBuf[4]	Time out setting, normal=100, unit=ms.
wBuf[5]	Not used
wBuf[6]	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive.

fBuf: Float Input/Output Table

Not used

■ DisableLinearMapping

Description:

Disable the linear mapping function for I-7014 or I-7016 module.

Syntax:

DisableLinearMapping (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]	COM port number, 1 to 255
wBuf[1]	Module address: 0x00 to 0xFF
wBuf[2]	Module ID: 0x7014, 0x7016
wBuf[3]	Checksum: 0=disable, 1=enable
wBuf[4]	Time out setting, normal=100, unit=ms.
wBuf[5]	Not used
wBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadLinearMappingStatus

Description:

Obtain the status of the linear mapping function for I-7014 or I-7016 module.

Syntax:

ReadLinearMappingStatus (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7014, 0x7016
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**0: linear mapping is disable**
	1: linear mapping is enable
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadSourceValueOfLM

Description:

Obtain the setting value of Linear Mapping source range for I-7014/7016 module.

Syntax:

ReadSourceValueOfLM (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	ORD Input/Output argument table
fBuf:	float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7014, 0x7016
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**Low Source Value**
fBuf[1]:	**High Source Value**

■ ReadTargetValueOfLM

Description:

Obtain the setting value of Linear Mapping target range for I-7014/7016 modules.

Syntax:

ReadTargetValueOfLM (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7014, 0x7016
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**Low Target Value**
fBuf[1]:	**High Target Value**

6.3 Analog Output Functions

6.3.1 I-7000 series modules

■ AnalogOut

Description:

Output the analog value from Analog output module of I-7000 series modules.

Syntax:

AnalogOut (WORD wBuf [], float fBuf [], char szSend [], char szReceive [])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, ID: 0x7016/21/22/24
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5]:	The analog output channel No. (0 to 3) of module I-7024; No used for single analog output module
wBuf[6]:	0 → no save to szSend & szReceive 1 → Save to szSend & szReceive.

fBuf: Float Input/Output Table

fBuf[0] :	Analog output value

■ AnalogOutReadBack

Description:

Read back the analog output value of analog output modules for I-7000 series modules. There are two types of read back functions, as described in the following:

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBack (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7016/7021/7022/7024
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0: command $AA6 read back 1: command $AA8 read back
wBuf[6]:	0 → no save to szSend & szReceive 1 → Save to szSend & szReceive.
wBuf[7] :	The analog output channel No. (0 to 3) of module I-7024 No used for single analog output module

fBuf: Float Input/Output Table

fBuf[0] :	**Analog output read back value**

■ AnalogOutHex

Description:

Output the analog value of analog output modules through Hex format.

Syntax:

AnalogOutHex (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7021/21P/22
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	The analog output channel No. (0 to 3) of module I-7024; No used for single analog output module
wBuf[6]:	0 → no save to szSend & szReceive 1 → save to szSend & szReceive
wBuf[7]:	Analog output value in Hexadecimal Data format

fBuf: Float Input/Output Table

Not used.

■ AnalogOutFsr

Description:

Output the analog value of analog output modules through % of span data format. This function only can be used after analog output module is set as “FSR” output mode.

Syntax:

AnalogOutFsr (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7021/21P/22
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	The analog output channel No. (0 to 3) of module I-7024; No used for single analog output module
wBuf[6]:	0→ no save to szSend & szReceive 1→ save to szSend & szReceive

fBuf: WORD Input/Output Table

fBuf[0]:	Analog output value in % of Span data format.

■ AnalogOutReadBackHex

Description:

Read back the analog output value of analog output modules in hex format for I-7000 series modules. There are two types of read back functions, as described in the following:

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBackHex (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
SzSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0];	COM port number: 1 to 255
wBuf[1];	Module address: 0x00 to 0xFF
wBuf[2];	Module ID: 0x7021/21P/22
wBuf[3];	Checksum: 0=disable, 1=enable
wBuf[4];	Time out setting, normal=100, unit=ms.
wBuf[5]:	0: command $AA6 read back
	1: command $AA8 read back
wBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7] :	The analog output channel No. (0 to 3) of module I-7024
	No used for single analog output module
dwBuf[9]	Analog output value in Hexadecimal Data format

fBuf: Float Input/Output Table

Not used.

■ AnalogOutReadBackFsr

Description:

Read back the analog output value of analog output modules through % of span data format for I-7000 series modules.

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBackFsr (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
SzSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0];	COM port number: 1 to 255
wBuf[1];	Module address: 0x00 to 0xFF
wBuf[2];	Module ID: 0x7021/21P/22
wBuf[3];	Checksum: 0=disable, 1=enable
wBuf[4];	Time out setting, normal=100, unit=ms.
wBuf[5]:	0: command $AA6 read back
	1: command $AA8 read back
wBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7] :	The analog output channel No. (0 to 3) of module I-7024
	No used for single analog output module
wBuf[9]	Analog output value in % of Span data format

fBuf: Float Input/Output Table

Not used.

6.3.1 I-8000 series modules

■ AnalogOut_8K

Description:

Output the analog value of analog output module for I-8000 series modules.

Syntax:

AnalogOut_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8024
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The defined analog output channel No.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit

fBuf: Float Input/Output Table

fBuf[0] :	Analog output value

■ AnalogOutReadBack_8K

Description:

Read back the analog output value of analog output module for I-8000 series modules. This function currently supports I-8022/8024/8026 modules.

Syntax:

AnalogOutReadBack_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2]:	Module ID, 0x8024
dwBuf[3]:	0=checksum disable, 1=checksum enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The defined analog output channel No.
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.

fBuf: Float Input/Output Table

fBuf[0] :	**Analog output read back value**

6.3.3 I-87K series modules

■ AnalogOut_87K

Description:

Output the analog value of analog output module for I-87K series modules.

Syntax:

AnalogOut_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87016/ 0x87021/0x87022/0x87024/0x87026
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	The defined analog output channel No.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0] :	Analog output value

■ AnalogOutReadBack_87K

Description:

Read back the analog output value of analog output modules for I-87K series modules. This function currently supports I-87021/22/24/26. There are two types of read back functions, as described in the following:

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBack_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address, from 0x00 to 0xFF
dwBuf[2]:	Module ID, 0X87016/87021/87024/87026
dwBuf[3]:	0=checksum disable, 1=checksum enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The defined analog output channel No
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0] :	**Analog output read back value**

■ AnalogOutHex_87K

Description:

Output the analog value of analog output I-87K series modules through Hex format.

Syntax:

AnalogOutHex_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address: 0x00 to 0xFF
dwBuf[2]:	Module ID: 0x87022/26
dwBuf[3]:	Checksum: 0=disable, 1=enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The analog output channel No. (0 to 1)
dwBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7]:	Analog output value in Hexadecimal Data format

fBuf: Float Input/Output Table

Not used.

■ AnalogOutFsr_87K

Description:

Output the analog value of analog output through % of span data format for I-87K series modules. This function only can be used after analog output module is set as “FSR” output mode.

Syntax:

AnalogOutFsr_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: WORD Input/Output Table

dwBuf[0]:	COM port number, 1 to 255
dwBuf[1]:	Module address: 0x00 to 0xFF
dwBuf[2]:	Module ID: 0x87022/87026
dwBuf[3]:	Checksum: 0=disable, 1=enable
dwBuf[4]:	Time out setting, normal=100, unit=ms.
dwBuf[5]:	The analog output channel No. (0 to 1)
dwBuf[6]:	0→ no save to szSend & szReceive
	1→ save to szSend & szReceive

fBuf: WORD Input/Output Table

fBuf[0]:	Analog output value in % of Span data format.

■ AnalogOutReadBackHex_87K

Description:

Read back the analog output value of analog output modules in hex format for I-87K series modules. There are two types of read back functions, as described in the following:

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBackHex_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
SzSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

dwBuf[0];	COM port number: 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87022/87026
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5]:	0: command $AA6 read back
	1: command $AA8 read back
dwBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	The analog output channel No. (0 to 1)
dwBuf[9]	**Analog output value read back in Hexadecimal Data format**

fBuf: Float Input/Output Table

Not used.

■ AnalogOutReadBackFsr_87K

Description:

Read back the analog output value of analog output modules through % of span data format for I-87K series modules. There are two types of read back functions, as described in the following:

1. Last value is read back by $AA6 command

2. Analog output of current path is read back by $AA8 command

Syntax:

AnalogOutReadBackFsr_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
SzSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: WORD Input/Output Table

dwBuf[0];	COM port number: 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87022/87026
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5]:	0: command $AA6 read back
	1: command $AA8 read back
dwBuf[6]	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	The analog output channel.

fBuf: Float Input/Output Table

fBuf[0]:	**Analog output value read back in % of Span data format.**

6.4 Digital Input Functions

6.4.1 I-7000 series modules

■ DigitalIn

Description:

Obtain the digital input value from I-7000 series modules.

Syntax:

DigitalIn (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7050/7052/7053/7060/7041/7044
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	**16-bit digital input data**
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalInLatch

Description:

Obtain the latch value of the high or low latch mode of Digital Input module.

Syntax:

DigitalInLatch (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/52/53/60/63/65/41/44
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0: low Latch mode , 1: high Latch mode
wBuf[6]:	0 → no save to szSend & szReceive
	1 →Save to szSend & szReceive
wBuf[7]:	**Latch value**

fBuf: Float Input/Output Table

Not used

■ ClearDigitalInLatch

Description:

This function can clear the latch status of digital input module when latch function has been enabled.

Syntax:

ClearDigitalInLatch (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/52/53/60/63/65/41/44
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalInCounterRead

Description:

Obtain the counter event value of the channel number of Digital Input module.

Syntax:

DigitalInCounterRead (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0];	COM port number, 1 to 255
wBuf[1];	Module address: 0x00 to 0xFF
wBuf[2];	Module ID: 0x7050/52/53/60/63/65/41/44
wBuf[3];	Checksum: 0=disable, 1=enable
wBuf[4];	Time out setting, normal=100, unit=ms.
wBuf[5];	The digital input Channel No.
wBuf[6];	0 → no save to szSend & szReceive 1 → Save to szSend & szReceive
wBuf[7]:	**Counter value of the digital input channel No.**

fBuf: Float Input/Output Table

Not used

■ ClearDigitalInCounter

Description:

Clear the counter value of the channel number of Digital Input module.

Syntax:

ClearDigitalInCounter (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]	Module ID: 0x7050/52/53/60/63/65/41/44
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	The digital input Channel No.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadEventCounter

Description:

Obtain the value of event counter of I-7000 series modules. This function only supports I-7011, I-7012, and I-7014 modules.

Syntax:

ReadEventCounter (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011/0x7012/0x7014/0x7016
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive
wBuf[7]:	**The value of event counter**

fBuf: Float Input/Output Table

Not used

■ ClearEventCounter

Description:

Clear the value of event counter of I-7000 series modules. This function only supports I-7011, I-7012, and I-7014 modules.

Syntax:

ClearEventCounter (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011/0x7012/0x7014/0x7016
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

6.4.2 I-8000 series modules

■ DigitalIn_8K

Description:

Obtain the digital input value from I-8000 series modules.

Syntax:

DigitalIn_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8040/42/51/52/53/54/55/58/63/77
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	**16-bit digital input data**
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.

fBuf: Float Input/Output Table

Not used

6.4.3 I-87K series modules

■ DigitalIn_87K

Description:

Obtain the digital input value from I-87K series modules.

Syntax:

DigitalIn_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87054/55/56/57/60/63/64/65/66/68
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	**16-bit digital input data**
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalInLatch_87K

Description:

Obtain the digital Input latch value of the high or low latch mode of I-87K series modules.

Syntax:

DigitalInLatch_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dWBuf: WORD Input/Output Table

dwBuf[0];	COM port number, 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87051/52/53/54/58/63
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5];	0: low latch mode, 1: high latch mode
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7];	**Latch value**

fBuf: Float Input/Output Table

Not used

■ ClearDigitalInLatch_87K

Description:

This function can clear the latch status of digital input module when latch function has been enabled.

Syntax:

ClearDigitalInLatch_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dWBuf: WORD Input/Output Table

dwBuf[0];	COM port number, 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87051/52/53/54/63
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5];	Not used
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalInCounterRead_87K

Description:

Obtain the counter value of the digital input channel No. of I-87K series modules.

Syntax:

DigitalInCounterRead_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dWBuf: WORD Input/Output Table

dwBuf[0];	COM port number, 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87051/52/53/54/63
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5];	The digital input Channel No.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7]:	**Counter value of the digital input channel No.**

fBuf: Float Input/Output Table

Not used

■ ClearDigitalInCounter_87K

Description:

Clear the counter value of the digital input channel No. of I-87K series modules.

Syntax:

ClearDigitalInRead_87K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dWBuf: WORD Input/Output Table

dwBuf[0];	COM port number, 1 to 255
dwBuf[1] ;	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x87051/52/53/54/63
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5];	The digital input channel No.
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

6.5 Digital Output Functions

6.5.1 I-7000 series modules

■ DigitalOut

Description:

Output the value of the digital output module for I-7000 series modules.

Syntax:

DigitalOut (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7050/60/63/65/66/67/42/43/44/11/12/14/80
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	16-bit digital output data
wBuf[6] :	0 → no save to szSend & szReceive
	1 →Save to szSend & szReceive.

fBuf: Float Input/Output Table

Not used

■ DigitalBitOut

Description:

Set the digital output value of the channel No. of I-7000 series modules. The output Value is “0” or “1”.

Syntax:

DigitalBitOut (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/60/63/65/66/67/42/43/44
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → Save to szSend & szReceive.
wBuf[7]:	The digital output channel No.
wBuf[8]:	Logic value (0 or 1)

fBuf: Float Input/Output Table

Not used

■ DigitalOutReadBack

Description:

Read back the digital output value of I-7000 series modules.

Syntax:

DigitalOutReadBack (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7050/60/66/67/42/43/44
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	**16-bit digital output data read back**
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalOut_7016

Description:

Set the digital output value of the specified channel No. of I-7016 module. If the parameter of wBuf[7] is “0”, it means to output the digital value through Bit0 and Bit1 digital output channels. If wBuf [7] is “1”, it means to output the digital value through Bit2 and Bit3 digital output channels

Syntax:

DigitalOut_7016 (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7016
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	2-bit digital output data in decimal format
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**0: Bit0, Bit1 output**
	1: Bit2, Bit3 output

fBuf: Float Input/Output Table

Not used

6.5.2 I-8000 series modules

■ DigitalOut_8K

Description:

Set the digital output value of digital output module for I-8000 series modules.

Syntax:

DigitalOut_8K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules..

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x8041/42/54/55/56/57/60/63/64/65/66/68/77
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	16-bit digital output data
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7] :	Slot number; the I/O module installed in I-8000 main unit.

fBuf: Float Input/Output Table

Not Used

■ DigitalBitOut_8K

Description:

Set the digital value of the digital output channel No. of I-8000 series modules. The output value is “0” or “1”

Syntax:

DigitalBitOut_8K (DWORD dwBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

dwBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-8000 series modules.
szReceive:	Result string receiving from I-8000 series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: WORD Input/Output Table

dwBuf[0];	COM port number, 1 to 255
dwBuf[1];	Module address: 0x00 to 0xFF
dwBuf[2];	Module ID: 0x8041/42/54/55/56/57/60/63/64/65/66/68/77
dwBuf[3];	Checksum: 0=disable, 1=enable
dwBuf[4];	Time out setting, normal=100, unit=ms.
dwBuf[5];	Output digital data; 0 or 1
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7];	Slot number; the I/O module installed in I-8000 main unit.
dwBuf[8];	The output channel No.

fBuf: Float Input/Output Table

Not used

6.5.3 I-87K series modules

■ DigitalOut_87K

Description:

Set the digital output value of the digital output module for I-87K series modules.

Syntax:

DigitalOut_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID0x87054/55/56/57/60/63/64/65/66/68
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	16-bit digital output data
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalOutReadBack_87K

Description:

Read back the digital output value of the digital output module for I-87K series modules.

Syntax:

DigitalOutReadBack_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: DWORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87054/55/56/57/60/63/64/65/66/68
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	**16-bit digital output data read back**
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DigitalBitOut_87K

Description:

Set the digital output value of the specific digital output channel No. of the digital output module for I-87K series modules. The output value is only for “0” or “1”

Syntax:

DigitalBitOut_87K (DWORD dwBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

dwBuf:	DWORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-87K series modules.
szReceive:	Result string receiving from I-87K series modules.

Return Value:

NoError:	OK
Others:	Error code

dwBuf: WORD Input/Output Table

dwBuf[0] :	COM port number, 1 to 255
dwBuf[1] :	Module address, from 0x00 to 0xFF
dwBuf[2] :	Module ID, 0x87054/55/56/57/60/63/64/65/66/68
dwBuf[3] :	0=checksum disable, 1=checksum enable
dwBuf[4] :	Time out setting, normal=100, unit=ms.
dwBuf[5] :	1-bit digital output data
dwBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
dwBuf[7]:	The digital output channel No.
dwBuf[8]:	Data to output (0 or 1)

fBuf: Float Input/Output Table

Not used

6.6 Counter Functions

■ CounterIn_7080

Description:

Obtain the value of the selected counter in module I-7080.

Syntax:

CounterIn_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → Set counter 0
	1 → Set counter 1
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceiveseries
wBuf[7]:	**High word of counter value from the selected counter**
wBuf[8]:	**Low word of counter value from the selected counter**

fBuf: Float Input/Output Table

Not used

■ StartCounting_7080

Description:

Start/stop the counting process of the selected counter in module I-7080.

Syntax:

StartCounting_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address, from 0x00 to 0xFF	
wBuf[2]:	Module ID, 0x7080	
wBuf[3]:	0=checksum disable, 1=checksum enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	0	→ Select the counter 0
	1	→ Select the counter 1
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive
wBuf[7]:	0	→ Stop Counting
	1	→ Start Counting

fBuf: Float Input/Output Table

Not used

■ ClearCounter_7080

Description:

Clear the value of the selected counter in module I-7080.

Syntax:

ClearCounter_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5] :	0 → Clear the value of counter 0
	1 → Clear the value of counter 1
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadCounterMaxValue_7080

Description:

Obtain the maximum setting value of the selected counter in module I-7080.

Syntax:

ReadCounterMaxValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → Select counter 0
	1 → Select counter 1
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**High word of maximum setting value from the selected counter**
wBuf[8]:	**Low word of maximum setting value from the selected counter**

fBuf: Float Input/Output Table

Not used

■ SetCounterMaxValue_7080

Description:

Configure the maximum value of the selected counter for module I-7080.

Syntax:

SetCounterMaxValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[],double MaxValue)

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.
MaxValue:	The maximum counter value

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address, from 0x00 to 0xFF	
wBuf[2]:	Module ID, 0x7080	
wBuf[3]:	0=checksum disable, 1=checksum enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	0	→ Configure counter 0
	1	→ Configure counter 1
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ EnableCounterAlarm_7080

Description:

Enable counter alarm (for alarm-mode 0) of I-7080 module.

Syntax:

EnableCounterAlarm_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7080
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	0 → enable alarm mode for counter 0
	1 → enable alarm mode for counter 1
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DisableCounterAlarm_7080

Description:

Disable the alarm mode of I-7080 module. This function only supports I-7080 module.

Syntax:

DisableCounterAlarm_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7080
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	Not used
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ EnableCounterAlarm_7080D

Description:

Enable the momentary alarm mode or latch alarm mode of I-7080D module. This function only supports I-7080D module.

Syntax:

EnableCounterAlarm_7080D (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7080
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	0 → momentary alarm mode
	1 → latch alarm mode
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ DisableCounterAlarm_7080D

Description:

Disable the alarm mode of I-7080D module. This function only supports I-7080D module.

Syntax:

DisableCounterAlarm_7080D (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255
wBuf[1] :	Module address, from 0x00 to 0xFF
wBuf[2] :	Module ID, 0x7080
wBuf[3] :	0=checksum disable, 1=checksum enable
wBuf[4] :	Time out setting, normal=100, unit=ms.
wBuf[5] :	Not used
wBuf[6] :	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadInputSignalMode_7080

Description:

Obtain the setting value of input signal mode in I-7080 module. For more detail information for “Input signal mode” please refer user’s manual.

Syntax:

ReadInputSignalMode_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255					
wBuf[1]:	Module address, from 0x00 to 0xFF					
wBuf[2]:	Module ID, 0x7080					
wBuf[3]:	0=checksum disable, 1=checksum enable					
wBuf[4]:	Time out setting, normal=100, unit=ms.					
wBuf[5]:	Not used					
wBuf[6]:	0	→ no save to szSend & szReceive				
	1	→ save to szSend & szReceive				
wBuf[7] :	**0**	**→ Counter:0**	**TTL**	**Counter:1**	**TTL**	
	1	**→ Counter:0**	**Photo**	**Counter:1**	**Photo**	
	2	**→ Counter:0**	**TTL**	**Counter:1**	**Photo**	
	3	**→ Counter:0**	**Photo**	**Counter:1**	**TTL**	

fBuf: Float Input/Output Table

Not used

■ SetInputSignalMode_7080

Description:

Configure the setting value of input signal mode in I-7080 module.

Syntax:

SetInputSignalMode_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255					
wBuf[1]:	Module address, from 0x00 to 0xFF					
wBuf[2]:	Module ID, 0x7080					
wBuf[3]:	0=checksum disable, 1=checksum enable					
wBuf[4]:	Time out setting, normal=100, unit=ms.					
wBuf[5] :	0	→ Counter:0	TTL	Counter:1	TTL	
	1	→ Counter:0	Photo	Counter:1	Photo	
	2	→ Counter:0	TTL	Counter:1	Photo	
	3	→ Counter:0	Photo	Counter:1	TTL	
wBuf[6]:	0	→ no save to szSend & szReceive				
	1	→ save to szSend & szReceive				

fBuf: Float Input/Output Table

Not used

■ PresetCounterValue_7080

Description:

Configure the preset value of the selected counter in I-7080 module.

Syntax:

PresetCounterValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[],double PresetValue)

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.
PresetValue:	The counter preset value

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address, from 0x00 to 0xFF	
wBuf[2]:	Module ID, 0x7080	
wBuf[3]:	0=checksum disable, 1=checksum enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	0	→ Configure counter 0
	1	→ Configure counter 1
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadPresetCounterValue_7080

Description:

Obtain the preset value of the selected counter in I-7080 module.

Syntax:

ReadPresetCounterValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → Read 7080's counter 0
	1 → Read 7080's counter 1
wBuf[6]:	0 → no save toSendTo7000&szReceive
	1 → save toSendTo7000&szReceive
wBuf[7]:	**The high word of preset value**
wBuf[8]:	**The low word of preset value**

fBuf: Float Input/Output Table

Not used

■ SetModuleMode_7080

Description:

Configure the alarm mode of the selected counter in I-7080 module. There are two counter alarm modes; alarm mode 0 and alarm mode 1. These two alarm modes can be used in both of I-7080 & I-7080D. For more detail information please refer user's manual.

Syntax:

SetModuleMode_7080 (WORD wBuf[], float fBuf[], char zSendTo7000[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → to set into 7080 alarm mode(mode 0)
	1 → to set into 7080D alarm mode (mode 1)
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadModuleMode_7080

Description:

Obtain the setting status of counter alarm mode in I-7080 module.

Syntax:

ReadModuleMode_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**0 → alarm mode 0**
	1 → alarm mode 1
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ SetLevelVolt_7080

Description:

Configure the high or low trigger level value of non-isolated input in I-7080 module.

Syntax:

SetLevelVolt_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → Set the low trigger level
	1 → Set the high trigger level
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	The trigger level value

■ ReadLevelVolt_7080

Description:

Obtain the high or low trigger level setting value of non-isolated input in I-7080 module.

Syntax:

ReadLevelVolt_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address, from 0x00 to 0xFF	
wBuf[2]:	Module ID, 0x7080	
wBuf[3]:	0=checksum disable, 1=checksum enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	0	→ Read the low trigger level
	1	→ Read the high trigger level
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**The trigger level setting value**

■ SetMinSignalWidth_7080

Description:

Configure the width value of the minimum high or low input signal level in I-7080 module.

Syntax:

SetMinSignalWidth_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[], long MinWidth)

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.
MinWidth:	The minimum input signal width of the high or low trigger level. The value unit is uS and the corresponding range is from 2 uS to 65535 uS. For Example: when MinWidth =2000, it means the minimum with is 2 mS.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → set the min. width at low level
	1 → set the min. width at high level
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadMinSignalWidth_7080

Description:

Obtain the setting width value of the minimum high or low input signal level in I-7080 module.

Syntax:

ReadMinSignalWidth_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0 → read the min. width at low level
	1 → read the min. width at high level
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**The input Signal Min Width**

fBuf: Float Input/Output Table

Not use

■ SetGateMode_7080

Description:

Configure the gate control mode of I-7080 module. There are 3 type modes:

0 → gate input signal must be low to enable counter
1 → gate input signal must be high to enable counter
2 → gate input signal is ignored. The counter will be always enable

Syntax:

SetGateMode_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address, from 0x00 to 0xFF	
wBuf[2]:	Module ID, 0x7080	
wBuf[3]:	0=checksum disable, 1=checksum enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	0	→ the gate is low active
	1	→ the gate is high active
	2	→ the gate is disable
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadGateMode_7080

Description:

Obtain the setting status of the gate control mode in I-7080 module.

Syntax:

ReadGateMode_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[5]:	**0 → gate control mode is low active**
	1 → gate control mode is high active
	2 → gate control mode is disable

fBuf: Float Input/Output Table

Not used

■ ReadOutputAlarmState_7080

Description:

Obtain the alarm digital output and the corresponding alarm setting status of I-7080 module.

Syntax:

ReadOutputAlarmState_7080 (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	No used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**For 7080 mode(alarm mode 0)**
	0 → Counter:0 disable Counter:1 disable
	1 → Counter:0 enable Counter:1 disable
	2 → Counter:0 disable Counter:1 enable
	3 → Counter:0 enable Counter:1 enable
	For 7080D mode (alarm mode 1)
	0 → Counter:0 disable
	1 → Counter:0 momentary alarm mode
	2 → Counter:0 latch alarm mode

Counter: 1 No used in Alarm mode 1

wBuf[8]: **Alarm digital output**

0	**→ DO:0**	**off**	**DO:1**	**off**
1	**→ DO:0**	**on**	**DO:1**	**off**
2	**→ DO:0**	**off**	**DO:1**	**on**
3	**→ DO:0**	**on**	**DO:1**	**on**

fBuf: Float Input/Output Table

Not used

■ ReadAlarmLimitValue_7080

Description:

Obtain the maximum value of the can read alarm limit value of I-7080 (for alarm-mode 0).

Syntax:

ReadAlarmLimitValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0];	COM port number, 1 to 255
wBuf[1];	Module address: 0x00 to 0xFF
wBuf[2];	Module ID: 0x7080
wBuf[3];	Checksum: 0=disable, 1=enable
wBuf[4];	Time out setting, normal=100, unit=ms.
wBuf[5];	0: Counter 0, 1: Counter 1
wBuf[6];	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]	**Hi-Word of counter value**
wBuf[8]	**Lo-Word of counter value**

fBuf: Float Input/Output Table

Not used

■ SetAlarmLimitValue_7080

Description:

Users can set alarm limit value for I-7080 (for alarm-mode 0).

Syntax:

SetAlarmLimitValue_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255	
wBuf[1]:	Module address: 0x00 to 0xFF	
wBuf[2]:	Module ID: 0x7080	
wBuf[3]:	Checksum: 0=disable, 1=enable	
wBuf[4]:	Time out setting, normal=100, unit=ms.	
wBuf[5]:	When in 7080 alarm mode(mode 0)	
	0:	To set Counter 0 alarm value
	1:	To set Counter 1 alarm value
	When in 7080D alarm mode (mode 0)	
	0:	To set Counter 0 high alarm value
	1:	To set Counter 0 high-high alarm value
wBuf[6]:	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadCounterStatus_7080

Description:

Obtain the counter working status (reading/stop) of I-7080 module.

Syntax:

ReadCounterStatus_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0] :	COM port number, 1 to 255	
wBuf[1] :	Module address, from 0x00 to 0xFF	
wBuf[2] :	Module ID, 0x7080	
wBuf[3] :	0=checksum disable, 1=checksum enable	
wBuf[4] :	Time out setting, normal=100, unit=ms.	
wBuf[5] :	0	→ to read Counter 0 status
	1	→ to read Counter 1 status
wBuf[6] :	0	→ no save to szSend & szReceive
	1	→ save to szSend & szReceive
wBuf[7]:	**0**	**→ Counting.**
	1	**→ Stop.**

fBuf: Float Input/Output Table

Not used

■ SetConfiguration_7080

Description:

Set the configuration of I-7080 or I-7080D module.

Syntax:

SetConfiguration_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

Notice

if you change the Baudrate or Checksum, please short the INIT*(pin6) to GND(pin10)

wBuf[0]	COM port number, 1 to 255	
wBuf[1];	Module original address: 0x00 to 0xFF	
wBuf[2];	Module ID: 0x7080	
wBuf[3];	Module original checksum:0=disable, 1=enable	
wBuf[4];	Time out setting, normal=100, unit=ms.	
wBuf[5];	Desired frequency gate time:	
		0: 0.1 second
		1: 1.0 second
	Don't care wBuf[5],if set the module in Counter mode	
wBuf[6];	Flag: 0=no save, 1=save send/receive string	
wBuf[7];	Desired new address	
wBuf[8];	Desired Type	1:Counter mode
		0:Frequency mode
wBuf[9];	Desired Baudrate:	

3: 1200 BPS 4: 2400 BPS

	5: 4800 BPS	6: 9600 BPS
	7: 19200 BPS	8: 38400 BPS
	9: 57600 BPS	10: 115200 BPS
wBuf[10];	Desired Checksum Address	

fBuf: Float Input/Output Table

Not used

■ DataToLed_7080

Description:

Output the defined data to LED display of I-7080D module.

Syntax:

DataToLed_7080 (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7080
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	Output data to LED display

6.7 Dual Watchdog Functions

All ICPDAS DCON (I-7000/8000/87K) series modules equip a hardware module watchdog and software host watchdog. The DCON series modules are designed for industry applications, therefore they can work in the harsh envioronment. About the detail “Dual Watchdog “ description please refer to “Appendix A”.

■ HostIsOK

Description:

This function provides a method to tell all module “Host PC is OK” by sending command string “~**”. If the module can’t receive “~**” during a time interval, the host “WatchDog” function will be enabled. The related functions are “ToSetupHostWatchdog” and “ToReadHostWatchdog”.

Syntax:

HostIsOK (WORD wBuf[], float fBuf[], char szSend [], char szReceive [])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Not used
wBuf[2]:	Not used
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ToSetupHostWatchdog

Description:

Configure the timer working interval of the Host Watchdog for I-7000 series modules. Also, it can enable or disable "WatchDog" function.

Syntax:

ToSetupHostWatchdog (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Not used
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	0: Disable host watchdog 1: Enable host watchdog If TimeOut value = 0 then disable the Host Watchdog
wBuf[6]:	0 → no save to szSend & szReceive 1 → save to szSend & szReceive
wBuf[7]:	Timer setting for watchdog, unit is 0.1 second, For example: when wBuf[7]=45, the timer interval is 4.5 second

fBuf: Float Input/Output Table

Not used

■ ToReadHostWatchdog

Description:

Obtain the setting value of timer interval and WatchDog status of the module for I-7000 series modules.

Syntax:

ToReadHostWatchdog (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Not used
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**0: Host watchdog is disabled.**
	1: Host watchdog is enabled.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**timer setting value for watchdog, unit is 0.1 second.**
	For example:
	When wBuf[7]=45, the timer interval is 4.5 second

fBuf: Float Input/Output Table

Not used

■ ReadModuleResetStatus

Description:

Obtain the module reset status. If the return value is “0”, it means “module has not been reset”. If the return value is “1”, it means “module has been reset”.

Syntax:

ReadModuleResetStatus (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID, 0x7011, 0x7012
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**0: module has not been reset**
	1: module has been reset
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadModuleHostWatchdogStatus

Description:

Obtain the module’s Host Watchdog status for I-7000 series modules. If the return value is “**0**”, it means “**Module’s Host watchdog is in NORMAL mode**”. If the return value is “**4**”, it means “**Module’s Host watchdog is in HOST FAILURE mode**”.

Syntax:

ReadModuleHostWatchdogStatus (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Not used
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**0: Module’s Host watchdog is in NORMAL mode.**
	4: Module’s Host watchdog is in HOST FAILURE mode.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ResetModuleHostWatchdogStatus

Description:

Reset the module's Host Watchdog status for I-7000 series modules. The related function is "ReadModuleHostWatchdogStatus".

Syntax:

ReadModuleHostWatchdogStatus (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Not used
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ SetSafeValueForDo

Description:

Configure the safe value of DO modules for I-7000 series modules when "WatchDog" function of the module has been enabled.

Syntax:

SetSafeValueForDo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/60/63/66/67/42/43/44
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Safe digital output value in Hex format
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ SetPowerOnValueForDo

Description:

Configure the initial digital output value of digital output module for I-7000 series modules when its power is on.

Syntax:

WORD SetPowerOnValueForDo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/60/63/66/67/42/43/44
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Power On value in Hex format
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ SetSafeValueForAo

Description:

Configure the channel No.safe value of analog output module for I-7000 series modules when the “WatchDog” function of the module has been enabled.

Syntax:

SetSafeValueForAo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7021/0x7024
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	The analog output channel No.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	Analog output safe value

■ SetPowerOnValueForAo

Description:

Configure the initial analog output of analog output module for I-7024 module when its power is on.

Syntax:

SetPowerOnValueForAo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7021/0x7024
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	The analog output Channel No
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	Power On analog output value

■ SetPowerOnSafeValue

Description:

Configure the power on and safe value of the digital output channels for I-7011, I-7012, and I-7014 modules.

Power On value:

00➔ DO0 off , DO1 off ;	01➔ DO0 on , DO1 off ;
02➔ DO0 off , DO1 on ;	03➔ DO0 on , DO1 on ;

Safe value:

00➔ DO0 off , DO1 off ;	01➔ DO0 on , DO1 off ;
02➔ DO0 off , DO1 on ;	03➔ DO0 on , DO1 on ;

Syntax:

SetPowerOnSafeValue (WORD wBuf[], float fBuf[], char zSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address, from 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7011/0x7012/0x7014
wBuf[3]:	0=checksum disable, 1=checksum enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Power On value in hex format
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	Safe value in hex format.

fBuf: Float Input/Output Table

Not used

■ ReadSafeValueForAo

Description:

Obtain the safe setting value of analog output module for I-7000 series modules.

Syntax:

ReadSafeValueForAo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7021/0x7022/0x7024
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Not used if module ID is 7021 Channel No (0 to 1) if module ID is 7022 Channel No (0 to 3) if module ID is 7024
wBuf[6]	0 → no save to szSend & szReceive 1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**Safe value**

■ ReadPowerOnValueForAo

Description:

Obtain the initial output setting value of analog output module for I-7000 series module when the power is on.

Syntax:

ReadPowerOnValueForAo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID:0x 7024
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	Channel No.(0 to 3) if module ID is 7024
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**The initial output value when the power is on.**

■ ReadPowerOnValueForDo

Description:

Obtain the initial output setting value of digital output module for I-7000 series module when the power is on.

Syntax:

ReadPowerOnValueForDo (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	module address: 0x00 to 0xFF
wBuf[2]:	module ID: 0x7050/60/63/65/66/67/42/43/44
wBuf[3]:	checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

fBuf[0]:	**Power on value in hex format**

■ ReadSafeValueForDo

Description:

Obtain the safe setting value of digital output module for I-7000 series modules when “WatchDog” function of the module has been enabled.

Syntax:

ReadSafeValueForDo (WORD wBuf[], float fBuf[], char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0]:	COM port number, 1 to 255
wBuf[1]:	Module address: 0x00 to 0xFF
wBuf[2]:	Module ID: 0x7050/60/63/65/66/67/42/43/44
wBuf[3]:	Checksum: 0=disable, 1=enable
wBuf[4]:	Time out setting, normal=100, unit=ms.
wBuf[5]:	**Safe Value in hex format**
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive

fBuf: Float Input/Output Table

Not used

■ ReadConfigStatus

Description:

Obtain the configuration status of the module for I-7000 series modules. For more detail information of the parameter please refer to the user's manual.

Syntax:

ReadConfigStatus (WORD wBuf[], float fBuf[],char szSend[], char szReceive[])

Input Parameter:

wBuf:	WORD Input/Output argument table
fBuf:	Float Input/Output argument table
szSend:	Command string to be sent to I-7000 series modules.
szReceive:	Result string receiving from I-7000 series modules.

Return Value:

NoError:	OK
Others:	Error code

wBuf: WORD Input/Output Table

wBuf[0];	COM port number, 1 to 255
wBuf[1];	Module address: 0x00 to 0xFF
wBuf[2];	Module ID (for all modules)
wBuf[3];	Checksum: 0=disable, 1=enable
wBuf[4];	Time out setting, normal=100, unit=ms.
wBuf[5];	Not used
wBuf[6]:	0 → no save to szSend & szReceive
	1 → save to szSend & szReceive
wBuf[7]:	**Module address**
wBuf[8]:	**Module Range Code**
wBuf[9]:	**Module baudrate**
wBuf[10]:	**Module data format**

fBuf: Float Input/Output Table

Not used

APPENDIX A WatchDog

Operation Principle

All ICPDAS DCON (I-7000/8000/87K) series modules equip a hardware module watchdog and software host watchdog. The DCON series modules are designed for industry applications, therefore they can work in the harsh envioronment. The modules may be down if its application environment is very bad and produces the noise effect to not be overcome by modules. Therefore, **the built-in hardware module watchdog can reset the module if it is down under the effect of too large noise signal** (refer to Figure 5). Sometimes even the host-PC may be down for hardware or software reasons. **The software host watchdog can monitor the status of host PC.** This **host watchdog** can be applied to two situations:(1) If the host PC is down, all the output of DCON series modules will go to their predefined safe states for safety protection reason (refer to Figure 1 to 3).; (2) If the RS-485 network is broken, all the host command can not send to remote modules. This is very dangerous in real world application. The DCON series output modules will force their output going to their predefined safe state for safety consideration if the host watchdog is active. Therefore, these dual watchdog features, module and host watchdog, will increase the reliability of system.

1. **Host watchdog**: (software): If the host is down, all modules output go to their predefined safe value. (refer to Figure 1 to 4).
2. **Module watchdog**: (hardware):If the module is down, module will reset itself and output go to safe value. (refer to Figure 5 to 6).

■ Host WatchDog

The host may be down under the effect of the following condition:

1.	Noise too large	→ make host hardware going error
2.	Software problem	→ make host going to the dead lock state
3.	Hardware problem	→ host hardware is damaged
4.	The RS485 network is broken	→ can't send out command to modules

The software host watchdog is designed to monitor the host computer. If the host computer is down, the output value of the modules will automatically go to their predefined safe states to avoid unpredictable damaged. Followings are the three methods for Host watchdog after module host watchdog is enabled.

(1) Host PC sends command "~**" to every module to notify that the Host PC is OK. If module host watchdog is enabled, this command "~**" must be sent to the module within the timeout period of watchdog timer and reset timer to mean that Host PC is OK.

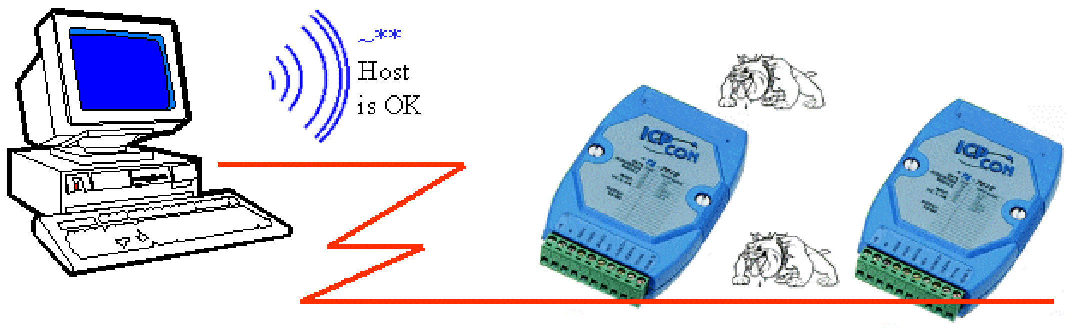


Figure 1. Host PC send command "~**" to every module to notify that the Host PC is OK.

(2) When the host PC command "~**" can not be sent to every module, which may be caused by host PC is down or RS-485 network is broken, that is, module watchdog timer can not be reset within default time period. Therefore,the module figures out the host PC is down and then it set the output value to the predefined safety value to avoid unpredictable damaged.

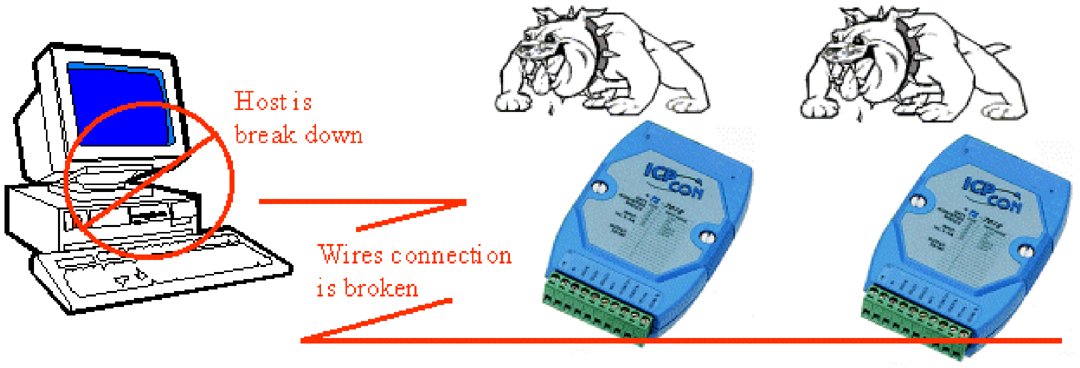


Figure 2. If Host PC can not send command to every module,then every Host Watchdog of the module set its output the predefined value for safety reason.

(3) Once the timer of host watchdog is not reset by host PC and module safety output is preduced, then any command will be ignored by the module. In addition, if PC is work and try to control the module again, host PC can send command "~AA0" to read in the module status and then use command " ~AA1" to clear the module status to let the module work again.

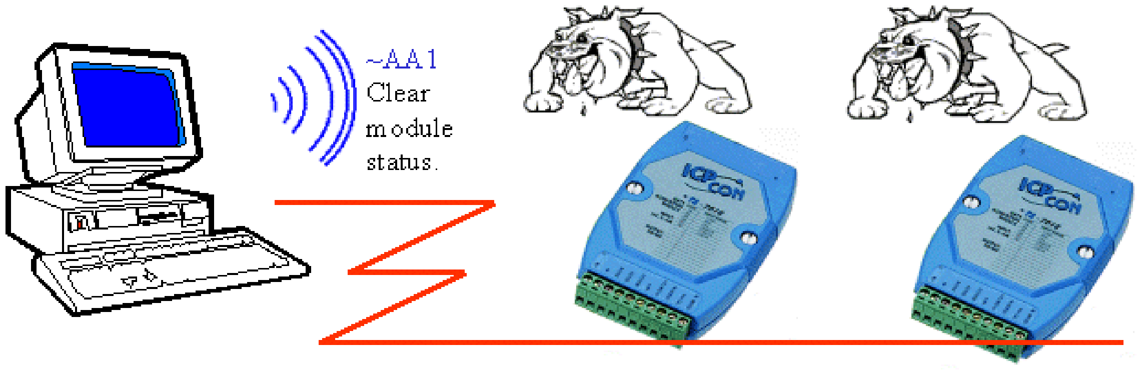


Figure 3. Host PC must use command ~AA0 to read the module status, and ~AA1 to clear the module status.

The flow chart of the host computer is given as Figure 4. Note that host OK cammand must be sent out to notify the module after every function sent.

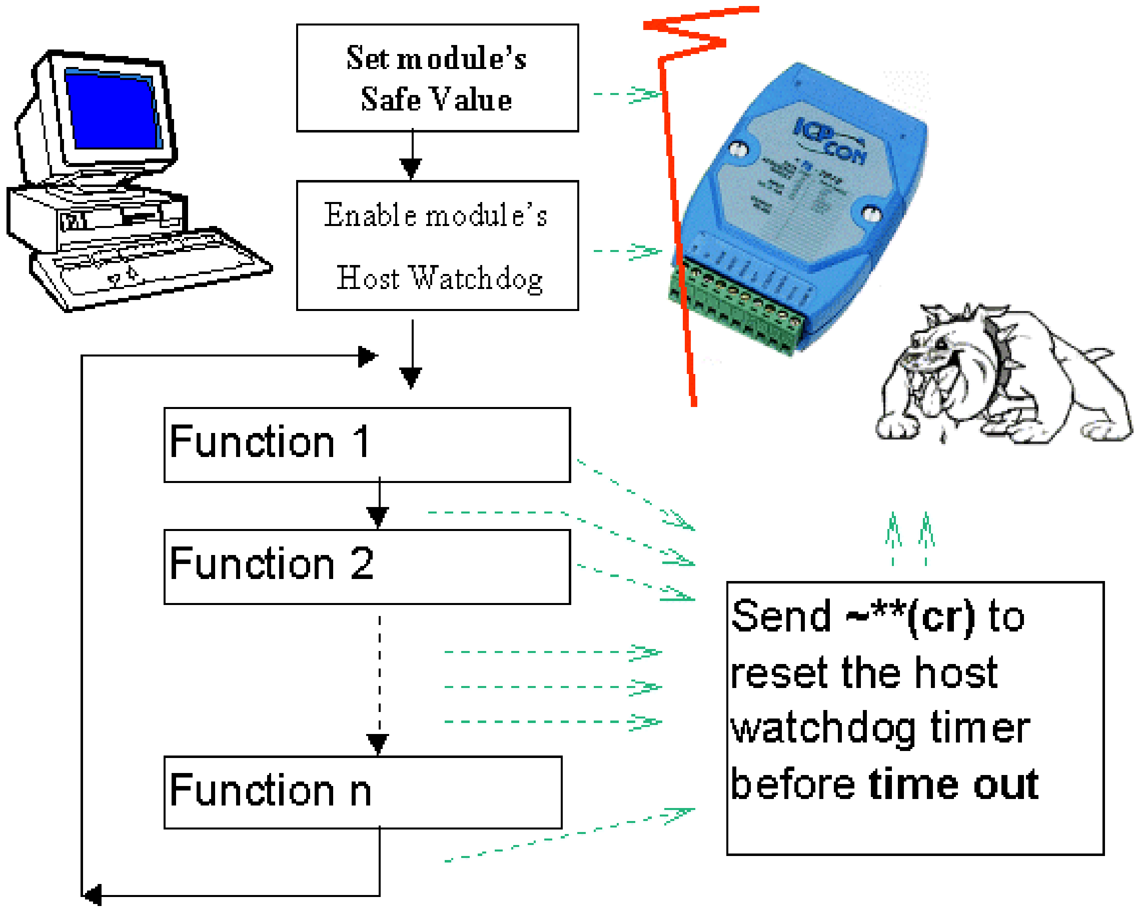

Set module's Safe Value
Enable module's Host Watchdog
Function 1
Function 2
Function n
Send ~**(cr) to reset the host watchdog timer before time out

Figure 4. The flow chart of Host Watchdog.

■ Module WatchDog

The module watchdog is a hardware reset circuit to monitor the modules's operation status. While working in a harsh or noisy environment, the module may be down by the external signal. The circuit may let the module to work continuous and never halt.

The reset status is set while the module power on or reset by module watchdog. And it can be cleared while the command ($AA5) of read reset status is applied. This command is useful for user to check the module working status. When the reset status is set means that the module is reset and the output may be change to the PowerOn value. When the reset status is clear means that the module is not reseted and the output is not change to PowerOn value. Note that the power on value can be set as different output value before module is reset. **Therefore, the user needs to send output command ($AA5) to module for checking and keeping the same output state before and after module watchdog reset.**

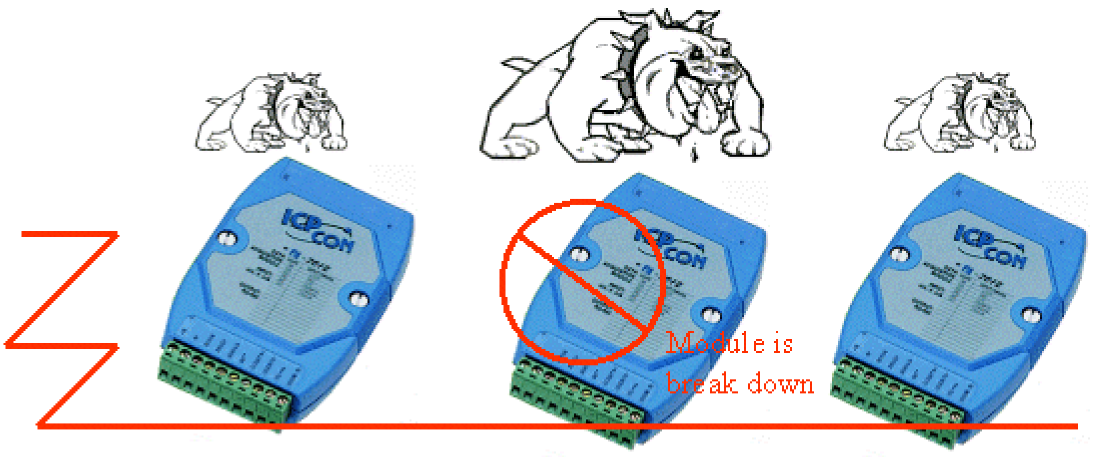


Figure 5. Module Watchdog will reset the module when the module is hanged.

The flow chart of the failure detection for module hardware watchdog is given as Figure 6.

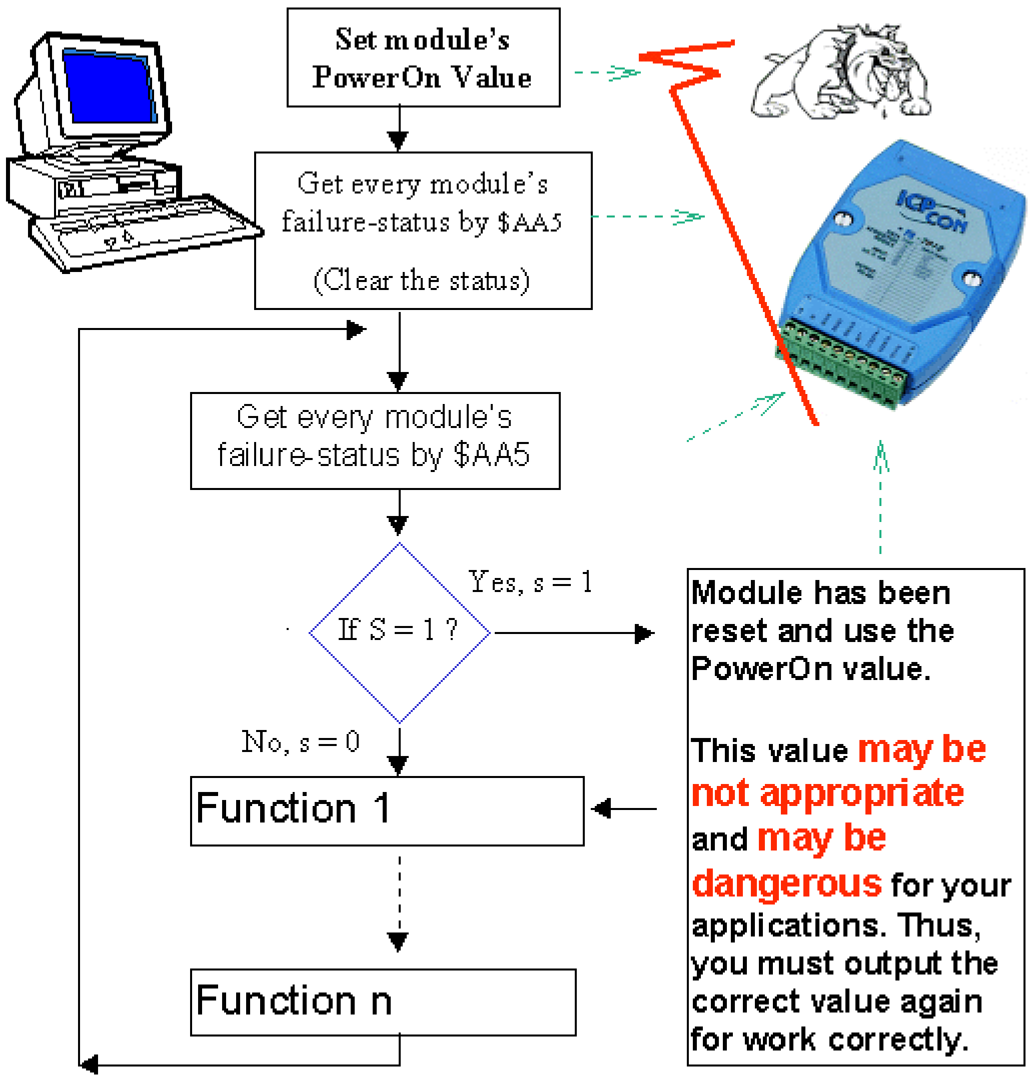

Set module's PowerOn Value
Get every module's failure-status by $AA5
(Clear the status)
Get every module's failure-status by $AA5
If S = 1 ?
Yes, s = 1
No, s = 0
Function 1
Function n
Module has been reset and use the PowerOn value.
This value may be not appropriate and may be dangerous for your applications. Thus, you must output the correct value again for work correctly.
ICP CON

Figure 6. The flow chart of Module Watchdog.

■ Comparison of Host and Module Watchdog

	Host Watchdog	Module Watchdog
Software or Hardware	● Software Watchdog ● Built-in firmware	● Hardware Watchdog ● Circuit in module
Purpose	● Monitor the Host PC ● Used in all output modules	● Monitor the Module ● Used in all modules
When to occur	● Host is down ● Communication line is broken	● Module is hanged ● Noise is too large in the working environment
What to do	● Module go to safe state ● Module status S = 0x04 ● Module's output go to safe value ● All output command will be ignored.	● Reset the Module ● Module Reset status S = 0x01 ● Module's output go to PowerOn value
CLEAR module-status	● ~AA1 ● S set to 0	
READ module-status	● ~AA0 ● S = 4 ➔ Host is down S = 0 ➔ Host is OK	
READ module-reset –status		● $AA5 ● S = 1 ➔ Module Reset ● S = 0 ➔ Not reset
Setup steps	● Setup the safe value ● Setup the timer interval value of Host Watchdog and enable the Host Watchdog	● Setup the PowerOn value
Send "Host is OK"	● ~** Send this command to modules before timeout of Host Watchdog's timer.	

APPENDIX B   Demo list

VISUAL C++ DEMO PROGRAM

These files are too large to print in this manual, refer to floppy disk for details.

\DCON_DLL\Demo\VC\

```
VC Demo List
|--\7k
|    |--\MFC
|    |    +--\Analog      ==> Analog Input demo
|    +--\SDK
|          |--\Demo01    ==> Send/Receive command with checksum disable
|          |--\Demo02    ==> Send/Receive command with checksum enable
|          |--\Demo03    ==> Send/Receive command to Counter, ANC PC-202
|          |--\Demo04    ==> Send/Receive command to OMRON PLC, CQM1 or
|          |                       C200H
|          |--\Demo05    ==> Multi-speed demo
|          |--\Demo06    ==> Multi-data-format demo
|          |--\Demo07    ==> Multi-speed & Multi-data-format demo
|          |--\Demo20    ==> Analog Input demo
|          |--\Demo21    ==> Analog Input demo
|          |--\Demo23    ==> Analog Output demo
|          |--\Demo24    ==> Digital Input demo
|          |--\Demo25    ==> Digital Output demo
|          |--\Demo26    ==> Multi-speed demo
|          |--\Demo27    ==> 8 Channels Analog Input demo
|          |--\Demo28    ==> Analog-Output ReadBack demo
|          +--\Demo29    ==> Digital-OutputReadBack demo
|--\87k
|    |--\Di              ==> Digital Input demo
|    |--\Do              ==> Digital Output and ReadBack demo
|    +--\DoBit           ==> Digital Bit-Output and ReadBack demo
+--\8k
      |--\Di             ==> Digital Input demo
      |--\Do             ==> Digital Output and ReadBack demo
      +--\DoBit          ==> Digital Bit-Output and ReadBack demo
```

DELPHI DEMO PROGRAM

These files are too large to print in this manual, refer to floppy disk for details.

\DCON_DLL\Demo\Delphi\

```
Delphi3 Demo List
|--\7k
|    |--\CompuTEX          ==> CompuTEX show demo
|    |--\UART
|    |      |--\Demo00      ==> Send/Receive Command demo
|    |      |--\Demo01      ==> Send_Receive_command demo
|    |      +--\SRCmd       ==> Send/Receive Command demo
|    |
|    |--\Analog
|    |      |--\Demo02      ==> Analog Input demo (for 7011/7012/7013/7014)
|    |      |--\Demo03      ==> Analog Input demo (for 7011/7012/7013/7014)
|    |      |--\Demo04      ==> Analog Input demo (for 7017/7018)
|    |      |--\Demo05      ==> Analog Output demo (for 7021)
|    |      |--\Demo06      ==> 8 Channels Analog Input demo
|    |      |--\Demo07      ==> Analog Input demo (for 7011/7012/7013/7014)
|    |      |--\Demo08      ==> Analog Output demo
|    |      |--\Demo09      ==> send synchronal command and read analog
|    |      |                   input value from several modules
|    |      |--\Demo10      ==> scan rate testing for analog input
|    |
|    |--\Digital
|    |      |--\Demo11      ==> Digital Input demo
|    |      |--\Demo12      ==> Digital Output demo
|    |      |--\Demo13      ==> Digital Input/Output demo (for 7050/7060)
|    |
|    |--\SafeAI         ==> safe value for AI module's output (for 7011/7012/7014)
|    |--\SafeAO         ==> safe value for AO module (for 7021/7024)
|    |--\SafeDO         ==> safe value for Digital output module
|    |
|    |--\Start1            ==> Getting Started (for 7011/7012/7014)
|    |--\WatchDog          ==> Dual Watchdog demo
|    +--\WatchDog2         ==> Dual WatchDog demo
|
|--\87k
```

```
|     |--\AnalogOut_87K       ==> AnalogOut_8K demo
|     |--\AnalogIn_87K        ==> AnalogIn_8K demo
|     |--\Di                  ==> Digital Input demo
|     |--\Do                  ==> Digital Output and ReadBack demo
|     +--\DoBit               ==> Digital Bit Output and ReadBack demo
|
|--\8k
|     |--\AnalogOut_8K        ==> AnalogOut_8K demo
|     |--\AnalogIn_8K         ==> AnalogIn_8K demo
|     |--\Di                  ==> Digital Input demo
|     |--\Do                  ==> Digital Output and ReadBack demo
|     +--\DoBit               ==> Digital Bit Output and ReadBack demo
```

VISUAL BASIC DEMO PROGRAM

These files are too large to print in this manual, refer to floppy disk for details.

\DCON_DLL\Demo\VB\

VB Demo List

```
|--\7k
|    |--\UART
|    |      |--\Demo00        ==> Send/Receive command demo (for 7000 series)
|    |
|    |--\Analog
|    |      |--\Demo01        ==> AnalogIn (for 7000 series)
|    |      |--\Demo02        ==> Analog Input demo (for 7017/7018)
|    |      |--\Demo03        ==> AnalogOut in "Hex" or "Fsr" format (for 7021)
|    |      |--\Demo04        ==> AnalogOut demo & performance evaluation
|    |      |--\Demo05        ==>AnalogInAll demo & performance evaluation
|    |      |--\Demo06        ==> Demo program for use MSComm and UART.DLL
|    |      |                     to read analog input value
|    |      |--\Demo07        ==> demo program for send synchronal command
|    |      |                     and read analog input value from several modules
|    |      |--\Demo08        ==> Demo program for 7016 and 7033 to set display
|    |      |                     channel of model and get display channel .
|    |      |--\Demo09        ==> Multi-speed demo (7012 and one 7021)
|    |
|    |--\Digital
|    |      |--\Demo10        ==> DigitalIn demo & performance evaluation
|    |      |--\Demo11        ==> DigitalOut demo & performance evaluation
|    |
|    |--\7080
|    |      |--\Demo12        ==> reading 7080/7080D Counter/Frequency value
|    |      |--\Demo13        ==> 7080/7080D Alarm demo
|    |--\Alarm                ==> Demo program for alarm of 7000 series module
|     |                         (Using I-7012 and I-7021 to simulate alarm mode)
|    |
|    |--\Strain Gauge
|    |      |--\Strain Gauge-1 ==> Stain Gauge (Using I-7016 to simulate
|    |      |   "LinearMap" function. Using I-7021 to simulate analog input source)
|    |      |
|    |      |--\Strain Gauge-1 ==> Application demo program of “Stain Gauge”.
|     |
|     |--\Demo08              ==> Multi-speed demo (7012 and one 7021)
|     |--\SafeAI              ==> Demo program for safe value of Analog input
|     |                           modules
|     |
|     |--\SafeAO              ==> Demo program for safe value for Analog output
|      |                          modules
|     |--\SafeDO              ==> Application of demo program for safe value for Digital
|     |                           Output modules
|     |--\Start               ==> demo program for Getting Start (for 7011/7012/7014)
|     |--\Watchdog            ==> Dual Watchdog for all modules (for 70xx)
|     +--\Watchdog2           ==> Dual Watchdog for all modules (for 70xx)
|
|--\87k
|     |--\DigitalIn_87K
|     |--\DigitalOut_87K
|     |--\DigitalBitOut_87K
|     |--\AnalogInAll_87K
|     +--\AnalogOut_87K & AnalogOutReadBack_87K
|
+--\8k
      |--\DigitalIn_8K
      |--\DigitalOut_8K
      |--\DigitalBitOut_8K
      |--\AnalogInAll_8K
      |--\AnalogOut_8K & AnalogOutReadBack_8K
      |--\AuotDemo            ==> DigitalOut_8K
      +--\InfoDemo            ==> Get module information
```

BORLAND C++ BUILDER DEMO PROGRAM

These files are too large to print in this manual, refer to floppy disk for details.

\DCON_DLL\Demo\BCB\

```
BCB3 Demo List
|--\7k
|    |--\UART
|    |      |--\Demo00        ==> Send/Receive command demo (for 7000 series)
|    |      |--\Demo01        ==> Send/Receive command demo (for 7000 series)
|    |--\Analog
|    |      |--\Demo02        ==> Analog Input demo (for 7011/7012/7013/7014)
|    |      |--\Demo03        ==> Analog Input demo (for 7011/7012/7013/7014)
|    |      |--\Demo04        ==> Analog Input demo (for 7017/7018)
|    |      |--\Demo05        ==> 8 Channels Analog Input demo (for 7017/7018)
|    |      |--\Demo06        ==> Analog Output demon (for 7021)
|    |      |--\Demo07        ==> Analog Output demo (for 7024)
|    |      |--\Demo08        ==> Synchronized Analog Input demo (for 7012)
|    |
|    |--\Digital
|    |      |--\Demo09        ==> Digital Input demo
|    |      |--\Demo10        ==> Digital Output demo
|    |      |--\Demo11        ==> 8 Channels Analog Input demo (for 7017/7018)
|    |      |--\Demo12        ==> Digital Input/Output demo (for 7050/7060)
|    |
|    |--\SafeAI         ==> safe value for AI module's output (for 7011/7012/7014)
|    |--\SafeAO         ==> safe value for AO module (for 7021/7024)
|    |--\SafeDO         ==> safe value for DO module
|    |--\Start1         ==> Getting Started (for 7011/7012/7014)
|    |--\Watchdog       ==> Dual Watchdog for all modules (for 70xx)
|    +--\Watchdog2      ==> Dual Watchdog for all modules (for 70xx)
|
|--\87k
|    |--\AnalogOut_8K         ==> AnalogOut_8K demo
|    |--\AnalogIn_8K          ==> AnalogIn_8K demo
|    |--\Di                   ==> Digital Input demo
|    |--\Do                   ==> Digital Output and ReadBack demo
|    +--\DoBit                ==> Digital Bit-Output and ReadBack demo
|
+--\8k
|       |--\AnalogOut_87K     ==> AnalogOut_87K demo
|       |--\AnalogIn_87K      ==> AnalogIn_87K demo
|       |--\Di                ==> Digital Input demo
|       |--\Do                ==> Digital Output and ReadBack demo
|       +--\DoBit             ==> Digital Bit-Output and ReadBack demo
```

APPENDIX C   Error Code

Error Code

Constant	Value	Description
NoError	0	Functions work normally.
FunctionError	1	Call wrong function error.
PortError	2	Use wrong COM Port error.
BaudRateError	3	Baud rate error.
DataError	4	Data Bit error.
StopError	5	Stop Bit error.
ParityError	6	Parity Bit error.
CheckSumError	7	CheckSum mechanism error.
ComPortNotOpen	8	COM is not open error.
SendThreadCreateError	9	Send thread create error.
SendCmdError	10	Send command error.
ReadComStatusError	11	Read COM Port status error.
ResultStrCheckError	12	Result string check error.
CmdError	13	Command error.
TimeOut	15	TimeOut error.
ModuleIdError	17	Module ID error.
AdChannelError	18	Channel number error.
UnderInputRange	19	Under input range error.
ExceedInputRange	20	Exceed input range error.
InvalidateCounterNo	21	Invalidate counter number error.
InvalidateCounterValue	22	Invalidate counter value error.
InvalidateGateMode	23	Invalidate gate mode error.
InvalidateChannelNo	24	Invalidate channel No error.
ComPortInUse	25	COM Port is in use error.

PROBLEMS REPORT

Technical support is available at no charge. The best way to report problems is send electronic mail to

service@icpdas.com

When reporting problems, please include the following information:

1. Is the problem reproducible? If so, how?
2. What kind and version of Platform are you using? For example, Windows 3.1, Windows for Workgroups, Windows NT 4.0, etc.
3. What kinds of our products are you using? Please see the product's manual.
4. If a dialog box with an error message was displayed, please include the full text of the dialog box, including the text in the title bar.
5. If the problem involves other programs or hardware devices, what devices or version of the failing programs do you use?
6. Other comments relative to this problem or any suggestions will be welcomed.

After we had received your comments, we will take about two business days to test the problems that you have reported. And then We will reply it as soon as possible to you. Please check that we had received your comments? And please keep in contact with us.

E-mail: service@icpdas.com
Web site: http://www.icpdas.com.tw